QUERY EXPANSION PADA LINE TODAY DENGAN ALGORITME *EXTENDED ROCCHIO RELEVANCE FEEDBACK*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Chandra Ayu Anindya Putri
NIM: 145150207111132

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

QUERY EXPANSION PADA LINE TODAY DENGAN
ALGORITME *EXTENDED ROCCHIO RELEVANCE FEEDBACK*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Chandra Ayu Anindya Putri
NIM: 145150207111132

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Januari 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Indriati, S.T, M.Kom	Dr.Eng Ahmad Afif Supianto, S.Si, M.Kom
NIP: 19831013 201504 2 002	NIK: 2012018206231001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Januari 2018

Chandra Ayu Anindya Putri

NIM: 145150207111132

KATA PENGANTAR

Puji syukur atas kehadirat Allah SWT atas segala karunianya yang telah melimpahkan rahmat, taufik, dan hidayah-Nya sehingga laporan penelitian skripsi ini yang berjudul "*Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback*" dapat terselesaikan dengan baik.

Melalui kesempatan ini, penulis menyadari penulisan skripsi ini tidak akan terselesaikan jika tanpa bantuan dari berbagai pihak. Oleh sebab itu, penulis ingin menyampaikan rasa hormat dan terimakasih yang sebesar-besarnya kepada segala pihak yang telah mendukung, memberikan bantuan, serta doa selama proses penulisan skripsi, diantaranya:

1. Ibu Indriati, S.T, M.Kom dan Bapak Dr.Eng Ahmad Afif Supianto, S.Si, M.Kom selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis dengan sabar sehingga penelitian skripsi ini dapat terselesaikan
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, Bapak Agus Wahyu Widodo, S.T, M.Cs dan Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku Ketua Jurusan, Ketua Program Studi dan Sekretaris Program Studi Teknik Informatika.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D., Bapak Ir. Heru Nurwarsito, M.Kom, Bapak Drs. Marji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan I, Wakil Dekan II dan Wakil Dekan III Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, Bapak Agus Wahyu Widodo, S.T, M.Cs dan Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku Ketua Jurusan, Ketua Program Studi dan Sekretaris Program Studi Teknik Informatika.
5. Ayahanda Djarwo Hawinoerrindrat, Ibunda Lusia Dewi Hayuningrum, kakak Chintya Dewi Ekacittasari, serta seluruh keluarga besar tercinta yang telah memberikan motivasi, kasih sayang, perhatian, serta senantiasa tiada hentinya memberikan doa demi kelancaran dan terselesaikannya skripsi ini.
6. Seluruh civitas akademik Teknik Informatika Universitas Brawijaya yang telah memberikan bantuan selama penulis menempuh studi dan selama penyelesaian skripsi di Teknik Informatika Universitas Brawijaya.
7. Teman saya tercinta, Putu Amelia Vennanda W., Nurul Muslimah, Riska Dewi Nurfarida, dan Nana Noviana, serta seluruh angkatan 2014 yang telah membantu dan memberikan dukungan selama proses penyelesaian penelitian skripsi ini.

8. Sahabat saya tercinta, Agustianamas Ciputra, Ananda Dwi Ariska, dan Dora Faizhatun yang senantiasa selalu mendukung dan memberikan doanya kepada saya demi kelancaran selama proses penyelesaian penelitian skripsi ini.

Penulis menyadari bahwa penyusunan skripsi ini masih memiliki banyak kekurangan, sehingga penulis membutuhkan adanya kritik maupun saran yang bersifat membangun. Akhir kata dari penulis, saya harap skripsi ini dapat memberikan manfaat bagi semua pihak yang menggunakannya.

Malang, 1 Januari 2018

Penulis

chandraayuanindyaputri@gmail.com

ABSTRAK

LINE TODAY memberikan akses informasi berupa konten-konten berita *up to date*. Data pada LINE TODAY dimanfaatkan untuk dapat dilakukan fitur pencarian berita. Teknik *Query Expansion* akan sangat berguna jika dikombinasikan dengan sistem pencarian, sebab *query* yang diinputkan pengguna akan dikombinasi dengan *query* tambahan yang diberikan oleh sistem. *Query* tambahan akan membuat *query* yang pengguna hasilkan lebih spesifik. Selain itu, hadirnya *feedback* pengguna (*user judgement/explicit relevance feedback*) yang melakukan penilaian pada tiap berita akan meminimalisir *query* yang ambigu. Proses yang dilakukan diawali dengan teknik *preprocessing*, yang terdiri dari beberapa tahapan, yaitu *cleansing, case folding, tokenization, filtering,* hingga *stemming.* Kemudian dilakukan pembobotan *term* (*term weighting*) dan *cosine similarity.* Setelah itu, proses yang dilakukan ialah perhitungan dengan metode *Extended Rocchio Relevance Feedback* yang merupakan metode turunan dari *Rocchio Relevance Feedback*, untuk menghasilkan *query* tambahan. Hasil yang diperoleh berdasarkan dari implementasi maupun pengujian pada penelitian *Query Expansion* pada LINE TODAY dengan Algoritme *Extended Rocchio Relevance Feedback* menghasilkan rata-rata nilai *Precision* sebesar 0.53308, *Recall* sebesar 0.81708, *F-Measure* sebesar 0.59553, dan Akurasi sebesar 0.9574. Nilai akurasi yang didapat dengan metode *Extended Rocchio Relevance Feedback* berdasar *user judgement* cenderung meningkat hingga 2% dibandingkan pencarian otomatis dengan metode *Rocchio Relevance Feedback.*

Kata kunci : *Text Mining, Query Expansion*, LINE TODAY, *Extended Rocchio Relevance Feedback.*

ABSTRACT

LINE TODAY provides access to up-to-date news contents. Data on LINE TODAY are used to be able to do search engine feature. Query Expansion technique will be very useful if it is to be combined with search engine system where the queries inputted by users are combined with additional queries from the system. These additional queries will make queries generated by users more specific. In addition, users feedback (user judgement/explicit relevance feedback) assessing on each news can minimize ambiguous queries. The process begins with preprocessing technique consisting of several stages which are cleansing, case folding, tokenization, filtering, and stemming. And then, term weighting and cosine similarity. The next process is calculated using the Extended Rocchio Relevance Feedback method which is a traditional method from Rocchio Relevance Feedback to generate an additional queries. The results are obtained from implementation and testing process of Query Expansion on LINE TODAY with Extended Rocchio Relevance Feedback Algorithm resulted an average Precision value of 0.53308, Recall value of 0.81708, F-Measure value of 0.59553, and Accuracy value of 0.9574. The accuracy value obtained with Extended Rocchio Relevance Feedback method based on user judgement increase by 2% compared to automated search by the method of Rocchio Relevance Feedback.

Key Words : Text Mining, Query Expansion, LINE TODAY, Extended Rocchio Relevance Feedback.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Informasi merupakan salah satu kebutuhan penting dan utama bagi seluruh masyarakat dari berbagai kalangan, guna untuk bertahan hidup. Pada era globalisasi ini, tak ada manusia yang tak luput dari kebutuhan informasi. Melalui informasi, kita dapat melakukan berbagai aktivitas atau kegiatan yang bermanfaat dan bahkan dapat menguntungkan. Tak hanya itu, informasi juga dapat membantu kita dalam pengambilan suatu keputusan tanpa merugikan pihak manapun, menambah wawasan, agar tahu apa yang sedang terjadi saat ini, serta meminimalisir asumsi-asumsi masyarakat yang belum dapat dipastikan kebenarannya. Informasi-informasi tersebut tentu didapatkan dengan melalui suatu perantara, yaitu media.

Saat ini, masyarakat telah masuk pada era globalisasi, dimana era tersebut membuat masyarakat mau tak mau diharuskan untuk memanfaatkan teknologi informasi berbasis internet atau *online*, karena penyampaian informasinya dapat disampaikan lebih cepat, mudah, dan lebih efektif. Dengan memanfaatkan internet, maka semakin cepat dan banyak pula informasi yang didapatkan, karena memiliki kapasitas dan cangkupan yang luas hingga ke seluruh penjuru dunia, seperti salah satu situs berita online, yaitu LINE TODAY. Situs berita tersebut merupakan salah satu fitur yang ada pada sebuah aplikasi pengiriman pesan atau *chat* bernama *LINE* dan menghadirkan berita-berita yang sedang hangat dibicarakan atau berita terkini. Meskipun berita yang tersedia pada LINE TODAY tidak berasal dari artikel pribadi melainkan dari sumber artikel lain, tapi sumber berita berasal dari sumber yang terpercaya. Penulis memanfaatkan data berupa dokumen berita dari LINE TODAY, karena mencoba menggunakan dokumen berbahasa Indonesia pada algoritme *Extended Rocchio Relevance Feedback*.

Permasalahan yang sering kita temui, ialah saat melakukan pencarian informasi pada mesin pencarian, *query* yang diinputkan terkadang tidak sesuai dengan hasil dokumen yang kita inginkan dan kurang spesifik sehingga kita akan kesusahan dalam menemukan dokumen yang ingin kita cari. Oleh sebab itu, dibutuhkan *query* baru atau tambahan *query* untuk membantu penyempurnaan inputan pada mesin pencarian. Berdasarkan permasalahan tersebut, maka diperlukan *query expansion* dengan menggunakan suatu metode yang dapat menampilkan *query* baru atau tambahan. Pada penelitian sebelumnya yang berjudul "*Query Expansion* Pada Sistem Temu Kembali Informasi Dokumen Berbahasa Indonesia Menggunakan *Pseudo Relevance Feedback*" (Pamungkas, Indriati, & Ridok, 2015), dilakukan pengujian dengan kata yang sesuai dengan *query* pertama sehingga *query* baru nantinya akan memiliki arti yang baru. Terbukti semakin banyak kata yang ditambahkan, maka *query* akan semakin spesifik dengan urutan dokumen yang semakin relevan. Pada penelitian sebelumnya yang berjudul "Pengembangan Sistem Penelusuran Katalog Perpustakaan Dengan Metode *Rocchio Relevance Feedback*" (Yugianus, Dachlan, & Hasanah, 2013), menyebutkan bahwa

pengimplementasian dengan metode *Rocchio Relevance Feedback* akan menampilkan hasil penelusuran dengan nilai kemiripan yang cukup tinggi.

Peneitian ini memanfaatkan eksplorasi metode yang pernah digunakan sebelumnya, yaitu pada metode *Rocchio Relevance Feedback*. Selain itu, juga memanfaatkan pendekatan *user judgement,* yaitu untuk menilai relevansi dokumen berdasar penilaian pengguna. Pada penelitian yang telah dilakukan sebelumnya yang berjudul, "Extending the Rocchio Relevance Feedback Algorithm to Provide Contextual Retrieval" (Jordan & Watters, 2004), pada penelitian tersebut dilakukan modifikasi *query* sehingga pencarian yang dilakukan nantinya akan lebih spesifik, karena dikombinasikan dari *query* yang kita masukkan dengan *query* tambahan dari sistem. Selain itu pada penelitian tersebut juga mendapatkan *feedback* dari pengguna untuk menilai relevansi tiap dokumennya.

Agar pencarian yang diberikan lebih efektif pada sistem IR, pada penelitian ini memanfaatkan model *Relevance Feedback.* Penelitian yang dilakukan sebelumnya juga membahas tentang beberapa metode lain yang digunakan pada *Relevance Feedback*. Penelitian tersebut berjudul, "*Relevance Feedback* Pada Temu-Kembali Teks Berbahasa Indonesia Dengan Metode *Ide-Dec-Hi* Dan *Ide-Regular*". Pada penelitian tersebut dilakukan pengimplementasian dan analisis pada kinerja perluasan *query* dengan beberapa metode dari *Relvance Feedback*, yaitu *Ide-Dec-Hi* dan *Ide-Regular*. Hasil penelitian menunjukkan adanya pengingkatan kinerja pada sistem temu kembali, karena dengan menggunakan metode *Ide-Dec-Hi* meningkat hingga 15.44% dan dengan menggunakan metode *Ide-Regular* peningkatan yang diperoleh mencapai 14.54%. Adanya penelitian tersebut telah terbukti bahwa penggunaan metode *Relevance Feedback* cocok untuk perluasan *query* (Adisantoso, Ridha, & Agusetyawan, 2006).

Pada penelitian ini, penulis melibatkan pengguna dalam perolehan relevansi dokumen yang didapatkan dari hasil *query* yang diberikan sehingga penulis memanfaatkan pendekatan *explicit relevance feedback/user judgement*. Seperti yang telah dilakukan pada penelitian sebelumnya, yaitu penelitian yang telah dilakukan oleh Saneifar, Hassan dkk, (2014). Pada penelitiannya yang berjudul, "Enhancing Passage Retrieval In Log Files By Query Expansion Based On Explicit And Pseudo Relevance Feedback", dilakukan dengan dua langkah, yaitu dengan *Explicit Relevance Feedback*, untuk mengidentifikasi konteks yang ada dalam informasi permintaan dan *Pseudo Relevance Feedback*. Kemudian langkah selanjutnya dilakukan pembobotan novel yang sesuai dengan *query* dan menggunakan *TRQ* (*Term Relatedness To Query*) dengan hasil penelitian, yaitu nilai *MRR (Mean Reciprocal Rank)* sebesar 87%. Dalam penelitian tersebut dapat diterapkan dengan baik pada file log maupun dokumen pada domain umum yang relevan (Saneifar, Bonniol, Poncelet, & Roche, 2014).

Berdasarkan beberapa penelitian di atas, penulis melakukan penelitian pada metode *Extended Rocchio Relevance Feedback*. Melalui pemanfaatan data pada LINE TODAY, penulis ingin membuat suatu mesin pencarian dengan tipe ekspansi *query* sehingga dapat memberikan *query* tambahan atau baru untuk membantu dan mempermudah pengguna dalam melakukan pencarian di situs berita online

tersebut. Oleh sebab itu, pada penelitian ini yang berjudul "*Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended* Rocchio Relevance Feedback", diharapkan mampu menampilkan hasil pencarian dokumen yang sesuai dengan apa yang diinputkan oleh pengguna.

1.2 Rumusan Masalah

Berdasar pada latar belakang yang telah dijelaskan sebelumnya, maka berikut merupakan rumusan masalah yang akan diangkat dalam permasalahan penelitian ini, diantaranya sebagai berikut:

1. Bagaimana penerapan *query expansion* dengan menggunakan metode *Extended Rocchio Relevance Feedback* pada pencarian berita online di LINE TODAY dapat memberikan *query* pencarian yang lebih spesifik?
2. Bagaimana hasil akurasi dan pengaruh dalam penggunaan *Query Expansion* dengan memanfaatkan metode *Extended Rocchio Relevance Feedback*?

1.3 Tujuan

Berikut merupakan tujuan yang diharapkan dari permasalahan yang ada pada penelitian, diantaranya sebagai berikut:

1. Menerapkan *query expansion* dengan menggunakan metode *Extended Rocchio Relevance Feedback* pada pencarian berita online di LINE TODAY dengan *query* pencarian yang lebih spesifik.
2. Mendapatkan hasil akurasi dan pengaruh pada *Query Expansion* yang berdasar dengan metode *Extended Rocchio Relevance Feedback*.

1.4 Manfaat

Manfaat yang diberikan dari penulisan pada penelitian ini diharapkan sebagai berikut:

1. Bagi Penulis

   Melalui penelitian ini penulis mendapatkan pemahaman dan mampu menerapkan *query expansion* dengan menggunakan metode *Extended Rocchio Relevance Feedback*.

2. Bagi Pengguna

   Melalui penelitian ini dengan memanfaatkan *query expansion*, pengguna dapat melakukan pencarian dengan mudah, karena adanya *query* baru yang akan membuat *query* pencarian menjadi lebih spesifik sehingga akan menampilkan dokmen yang diinginkan.

3. Bagi LINE TODAY

   Melalui penelitian ini dapat ditambahkan fitur baru, yaitu pencarian berita yang memanfaatkan model *query expansion* sehingga mempermudah pengguna dalam melakukan pencarian berita.

1.5 Batasan Masalah

Batasan masalah ini digunakan untuk meminimalisir permasalahan yang ada pada penelitian ini, berikut merupakan batasan masalahny:

1. Sistem yang dibuat berdasar pada penilaian pengguna.
2. Data yang digunakan berupa dokumen berita dari situs berita online LINE TODAY, yaiu sebanyak 200 data latih dan 25 data uji berupa *query*.
3. Metode *Query Expansion* yang digunakan ialah metode *Extended Rocchio Relevance Feedback*.
4. Sistem tidak memperhatikan kesalahan ejaan pada kata.
5. Sistem yang dibangun dengan memanfaatkan Bahasa Pemrograman PHP.

1.6 Sistematika Pembahasan

Sistematika penyusunan pada penulisan penelitian ini, disusun sebagai berikut:

BAB 1: PENDAHULUAN

Pada bab pendahuluan ini menjelaskan tentang latar belakang permasalahan, rumusan masalah, tujuan, manfaat, batasan masalah, hingga sistematika pembahasan pada penelitian *Query Expansion* Pada LINE TODAY Dengan Algoritme Extended *Rocchio Relevance Feedback*.

BAB 2: LANDASAN KEPUSTAKAAN

Pada bab landasan kepustakaan ini menjelaskan tentang teori, konsep, dan model yang digunakan untuk penelitian, serta adanya kajian pustaka dari penelitian sebelumnya yang digunakan sebagai referensi landasan dalam pembuatan skripsi ini.

BAB 3: METODOLOGI PENELITIAN

Pada bab metodologi ini membahas tentang metode dan teknik yang digunakan dalam penyelesaian suatu masalah, serta terdapat studi literature yang dimanfaatkan. Penyusunan bab ini terdiri dari studi literatur, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan analisis hasil pengujian, serta kesimpulan dan saran.

BAB 4: PERANCANGAN DAN IMPLEMENTASI

Pada bab ini membahas tentang hasil perancangan dan implementasi pada sistem dengan penyusunannya terdiri deskripsi sistem, manualisasi, perancangan sistem, hingga implementasi pada sistem.

BAB 5: PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis ini membahas tentang hasil pengujian yang dilakukan pada sistem, serta dilakukan analisis pada sistem yang telah dilakukan pengimplementasian.

BAB 6: PENUTUP

Pada bab penutup ini akan membahas jawaban kesimpulan yang ada pada subbab rumusan masalah, serta diberikan saran dari penulis untuk penelitian lebih lanjut. Penyusunannya terdiri dari kesimpulan dan saran.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada bab ini dilakukan pembahasan kajian pustaka sebagai acuan dalam pengerjaan skripsi yang didasarkan pada penelitian sebelumnya, serta berkaitan dengan *Query Expansion* yang memanfaatkan metode *Extended Rocchio Relevance Feedback*. Pada penelitian ini menggunakan beberapa kajian pustaka yang berasal dari beberapa jurnal penelitian, guna untuk memperkuat dalam pemecahan masalah penelitian yang akan dilakukan. Penelitian sebelumnya yang berjudul, "*Query Expansion* Pada Sistem Temu Kembali Informasi Dokumen Berbahasa Indonesia Menggunakan *Pseudo Relevance Feedback*", dilakukan penelitian pada model pencarian atau *query expansion* dengan metode yang digunakan ialah *Rocchio Relevance Feedback*, untuk menentukan kata atau *term* dan dengan pendekatan *Pseudo Relevance Feedback,* yang digunakan untuk mengambil dokumen teratas secara otomatis. Hasil pengujian akurasi yang didapatkan, ialah dengan menggunakan *Pseudo Relevance Feedback* sebesar 17% (Pamungkas, Indriati, & Ridok, 2015).

Penelitian lain yang dijadikan kajian pustaka, ialah penelitian sebelumnya yang berjudul "Pengembangan SIstem Penelusuran Katalog Perpustakaan Dengan Metode *Rocchio Relevance Feedback*", yang dilakukan oleh Pausta Yugianus dkk, (2013). Pada penelitian tersebut dilakukan sistem penelusuran pada katalog perpustakaan berbasis web dengan melakukan penentuan pada proses penelusuran berdasar input, yaitu *term* atau kata dari pengguna. Dari hasil penelitian tersebut dapat disimpulkan bahwa metode yang digunakan, yaitu *Rocchio Relevance Feedback*, mempermudah pencarian pada pustaka dan dapat menampilkan hasil dengan nilai kemiripan tertinggi, yaitu terletak pada yang sesuai dengan inputan *term* atau kata dari pengguna (Yugianus, Dachlan, & Hasanah, 2013).

Jurnal penelitian lainnya yang dijadikan kajian pustaka, ialah penelitian yang telah dilakukan oleh Saneifar, Hassan dkk, (2014). Pada penelitiannya yang berjudul, "Enhancing Passage Retrieval In Log Files By Query Expansion Based On Explicit And Pseudo Relevance Feedback", dimana pada penelitian tersebut dilakukan dengan dua langkah, yaitu dengan *Explicit Relevance Feedback*, untuk mengidentifikasi konteks yang ada dalam informasi permintaan dan *Pseudo Relevance Feedback* dengan hasil penelitian, yaitu nilai *MRR (Mean Reciprocal Rank)* sebesar 87%. Dalam penelitian tersebut dapat diterapkan dengan baik pada file log maupun dokumen pada domain umum yang relevan (Saneifar, Bonniol, Poncelet, & Roche, 2014).

Penelitian ini menggunakan metode *Extended Rocchio Relevance Feedback*, dimana pada penelitian sebelumnya, telah dilakukan oleh Jordan, Chris & Watters, Carolyn, (2004). Pada penelitiannya yang berjudul, "Extending the Rocchio Relevance Feedback Algorithm to Provide Contextual Retrieval", terfokus pada modifikasi metode *Rocchio Relevance Feedback,* yaitu *Extending Rocchio* yang

menunjukkan kinerja dokumen VSR dan hasil yang sebanding dengan algoritme *Rocchio Relevance Feedback.* Selain itu, pada penelitian tersebut penilaian untuk dokumen relevan maupun tidak relevan memanfaatkan *feedback* dari pengguna atau *user* (Jordan & Watters, 2004).

Penelitian yang dilakukan sebelumnya juga membahas tentang beberapa metode lain yang digunakan pada *Relevance Feedback*. Penelitian tersebut berjudul, "*Relevance Feedback* Pada Temu-Kembali Teks Berbahasa Indonesia Dengan Metode *Ide-Dec-Hi* Dan *Ide-Regular*". Pada penelitian tersebut dilakukan pengimplementasian dan analisis pada kinerja perluasan *query* dengan beberapa metode dari *Relvance Feedback*, yaitu *Ide-Dec-Hi* dan *Ide-Regular*. Hasil penelitian tersebut menunjukkan adanya pengingkatan kinerja pada sistem temu kembali, karena dengan menggunakan metode *Ide-Dec-Hi* meningkat hingga 15.44% dan dengan menggunakan metode *Ide-Regular* peningkatan yang diperoleh mencapai 14.54%. Dengan adanya penelitian tersebut telah terbukti bahwa penggunaan metode *Relevance Feedback* cocok untuk perluasan *query*, (Adisantoso, Ridha, & Agusetyawan, 2006).

Kajian pustaka di atas merupakan kajian-kajian yang dijadikan referensi atau rujukan oleh penulis dalam proses pengerjaan skripsi ini. Pada penelitian ini nantinya akan dilakukan pengimplementasian *Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback*.

2.2 *Text Mining*

Text mining, merupakan salah satu teknik untuk melakukan proses klasifikasi yang merupakan variasi dari data mining, guna untuk menemukan pola dari sekumpulan data tekstual untuk jumlah yang cukup besar. Selain untuk klasifikasi, dapat juga dimanfaatkan untuk menangani suatu masalah, seperti *clustering, information extraction,* dan *information retrieval*. Berikut merupakan langkah-langkah dalam proses *text mining* (Kurniawan, Effendi, & Sitompul, 2012), diantaranya:

a. *Text Processing*

Tahapan ini dilakukan *toLowerCase*, dimana dilakukan perubahan seluruh karakter dengan huruf kecil. Sedangkan tokenisasi untuk penguraian kalimat menjadi kata dan dihilangkan delimiternya.

b. *Feature Selection*

Tahapan ini dilakukan dengan menghilangkan *stopword* dan melakukan proses pemetaan dan penguraian dari suatu kata yang berimbuhan menjadi kata dasar.

2.3 LINE TODAY

Line merupakan aplikasi pengirim pesan online yang dapat diakses secara gratis melalui smartphone, tablet, hingga PC. Selain memudahkan kita dalam pengiriman pesan, Line juga dapat dengan mudah digunakan untuk melakukan *video call*, *voice call*, pengiriman foto, video, dokumen, pesan suara, dan lain

sebagainya. Pada fitur Line juga disediakan tempat untuk pembelian tema dan stiker yang biasa digunakan untuk emoticon. Tema maupun stiker yang tersedia bisa dibeli dengan menggunakan point yang kita miliki, dimana point bisa diisi dengan melakukan pembayaran, namun juga tersedia beberapa tema maupun stiker gratis tanpa membutuhkan point.

Fitur yang tak kalah penting dari Line adalah adanya fitur LINE TODAY, dimana fitur tersebut sangat membantu kita untuk mendapatkan informasi berita terkini dan ter-*update* yang diperbarui secara *real time*. Berita yang disuguhkan berupa berita-berita global yang tak hanya dalam negeri saja, namun juga mencangkup hingga berita luar negeri. Berita yang dihadirkan berasal dari artikel web berbagai mitra media yang terpercaya. Pada artikel-artikel yang tersedia, juga disediakan tempat untuk memberikan penilaian berupa "*like"* dan juga dapat memberikan komentar, namun kita harus memiliki akun line terlebih dahulu jika ingin mengaksesnyapada web browser di PC. Kita juga dapat mendapatkan pemberitahuan secara otomatis pada berita yang baru di *post*, yaitu dengan menambahkan pertemanan pada akun LINE TODAY, maka update berita akan dikirimkan pada kita berupa pesan.

2.4 Dasar Teori

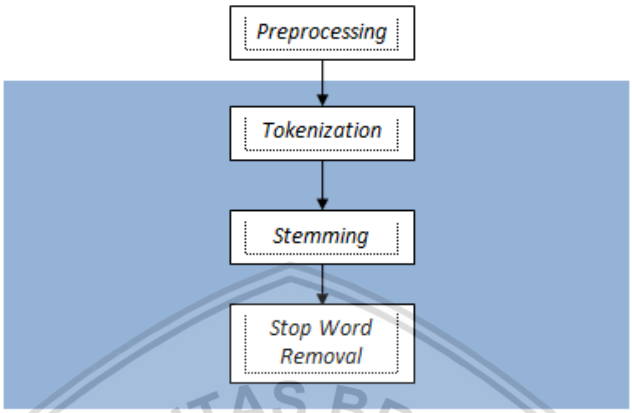
2.4.1 *Information Retrieval*

Information retrieval, merupakan sistem yang digunakan untuk merepresentasikan, melakukan penyimpanan, mengorganisasikan, serta memperoleh informasi (Yates & Neto, 1999). Informasi-informasi atau dokumen yang ditemukan pada keadaan yang tidak terstruktur atau berbentuk teks akan digunakan untuk memenuhi kebutuhan informasi dalam suatu koleksi yang besar (Manning, Raghavan, & Schütze, 2008). Dokumen yang dihasilkan dari *information retrieval* berupa daftar dokumen relevan dengan *inputan* yang diberikan oleh pengguna, yaitu dengan melalui pembanding antara *query* dari pengguna dengan *index* yang diberikan dari dokumen pada *information retrieval* (Yugianus, Dachlan, & Hasanah, 2013). Kata kunci atau *keyword* pada tiap dokumen atau *index term* yang diberikan, merupakan kata semantic yang membantu pengguna untuk mengetahui tema utama dari suatu dokumen (Yugianus, Dachlan, & Hasanah, 2013).

Information retrieval memiliki kemungkinan penggunaan istilah indeks yang digunakan untuk mewakili konten dari suatu dokumen, yang disebut dengan *unlimited aliasing* (Blair, David C., 2003, mengutip dari Furnas, Landauer, Gomez, & Dumais, 1987). Strategi dengan mengunakan *unlimited aliasing*, mengabaikan 2 hal, yaitu tidak ada batas jumlah kata maupun frase yang mewakili suatu konten dari informasi dan beberapa istilah indeks yang masuk ke dalam suatu dokumen, memiliki kemungkinan melakukan pengindeksan dengan kurang baik dari yang lainnya (Blair, David C., 2003, mengutip dari Brooks, 1993).

2.4.2 Pemrosesan Teks

Pada tahapan pemrosesan teks ini atau yang biasa dikenal dengan istilah *preprocessing,* dilakukan untuk mempermudah proses *information retrieval*, dimana kata yang dipilih akan diindekskan sebagai kata yang mewakili suatu dokumen. Berikut merupakan ilustrasinya (Vijayarani & Janani, 2016):



Gambar 2.1 Ilustrasi Proses *Preprocessing*

Pada gambar di atas menunjukan proses dari *preprocessing,* dimana terdiri dari proses *tokenization, stemming,* dan *stop word removal*. Berikut merupakan penjelasan proses *preprocessing* secara lengkap yang digunakan dalam penelitian ini, diantaranya:

2.4.2.1 *Cleansing dan Case Folding*

Proses *case folding* dilakukan pada semua huruf yang akan diubah menjadi huruf kecil atau *lowercase.* Sedangkan pada proses *cleansing*, dilakukan penghapusan pada komponen yang tidak dibutuhkan, seperti tag URL, hingga karakter lainnya yang ada dalam dokumen (Rustiana & Rahayu, 2017).

2.4.2.2 *Tokenization*

Proses ini melakukan pemecahan teks menjadi kata, istilah, hingga simbol, serta beberapa elemen penting lainnya. *Tokenizing* dilakukan pada kata kunci yang kemudian dilakukan suatu proses untuk diubah menjadi unit atau *token*, dimana unit tersebut biasanya berupa kata, angka, hingga tanda baca (Yugianus, Dachlan, & Hasanah, 2013). Proses ini pada umumnya terjadi pada level kata, namun tidak mudah dalam pendefinisian kata yang dimaksudkan (Vijayarani & Janani, 2016).

2.4.2.3 *Filtering*

Pada tahapan ini dilakukan dengan mengambil kata-kata atau *term* penting dari hasil proses sebelumnya, yaitu *tokenizing*. Kata-kata penghubung, seperti “yang”, “ke-“, “dari”, dan lain sebagainya akan dihilangkan. Pada proses ini memerlukan *stopword removal* untuk menghapus kata yang dianggap tidak penting (Yugianus, Dachlan, & Hasanah, 2013).

2.4.2.4 *Stemming*

Proses ini digunakan untuk mengurangi kata yang yang termodulasi pada suatu kata stem, root, hingga base (Vijayarani & Janani, 2016). Dalam proses ini pada teks berbahasa Indonesia perlu dilakukan sebelum proses *text mining*, karena memiliki *prefixes, suffixes, infexes* dan *confixes* yang membuat kata dasar dapat berubah menjadi lebih banyak bentuk sehingga pencarian yang dilakukan dengan kata dasar akan menjadi lebih sulit (Nata & Yudiastra, 2017).

Dalam proses *stemming* berbahasa Indonesia dapat dengan memanfaatkan dua algoritme, yaitu Nazief-Adriani dan Arifin-Setiono. Pada skripsi ini, akan memanfaatkan proses *stemming* dengan memanfaatkan algoritme Nazief-Adriani, dimana kamus akan dilakukan pengecekan setiap penerapan pada aturan *stemming* yang berhasil diidentifikasi, serta bila proses *stemming* berhasil menemukan akar kata, maka algoritma akan mengembalikan kata ke dalam kamus dan algoritma akan berhenti (Hamzah, Amir, 2006). Berikut merupakan langkah-langkah *stemming* dengan algoritme Nazief-Adriani (Wahyudi, Susyanto, & Nugroho, 2017):

1. Kata yang belum dilakukan *stemming* dicari di dalam kamus, jika tersedia, maka kata tersebut akan dianggap sebagai kata dasar yang tepat dan algoritma dihentikan.
2. Menghilangkan imbuhan infeksi atau *inflectional suffixes* (“-lah”,”-kah”,”-tah”, serta “-pun”). Kemudian jika berhasil dan akhirannya memiliki imbuhan (“-lah” atau ”-kah”), maka akan dilanjutkan ke langkah berikutnya dengan menghilangkan *inflectional posseive pronoun suffixes* (“-ku”,”-mu”, dan “-nya”). Cek apakah kata berada di dalam kamus kata dasar, jika ada, maka algoritma dihentikan, jika tidak, dilanjutkan ke step berikutnya.
3. Menghilangkan *derivational suffix*, yaitu (“-i” atau “-an”). Jika langkah tersebut berhasil, kemudian dilanjutkan ke langkah berikutnya, namun jika tidak, maka dilakukan hal berikut ini, diantaranya:
   a. Jika “-an” dihilangkan dan huruf terakir dari kata “-k”, maka “-k” juga dihilangkan dan dilanjutkan ke langkah berikutnya.
   b. Penghapusan akhiran (“-l”, “-an”, serta “-kan”) dikembalikan dan lanjut ke step berikutnya.
4. Penghapusan pada *derivational prefix* (“be-“,”di-“,”ke-“,”me-“,”pe-“,”se-“, dan ”te-“). Bila kata yang dimiliki tersedia dalam kamus kata dasar, maka proses akan dihentikan dan bila tidak tersedia, maka akan dilakukan *recoding.* Tahap-tahap proses dihentikan, karena memenuhi beberapa kondisi berikut ini, diantaranya:
   a Ada kombinasi awalan maupun akhiran yang tidak diijinkan
   b Awalan yang terdeteksi sama dengan awalan yang dihapuskan sebelumnya.
   c Tiga awalan dihilangkan.
5. Jika semua langkah sebelumnya telah dilakukan namun kata dasar belum ditemukan pada kamus, maka algoritma ini akan mengembalikan kata yang asli sebelum dilakukannya proses *stemming.*

2.4.3 *Term Weighting* (TF.IDF)

Perangkingan dokumen dengan menggunakan model VSM (*Vector Space Model*) direpresentasikan ke dalam *matriks* yang berisi bobot kata atau *term* pada suatu dokumen, diamana bobot dinyatakan sebagai kepentingan kata yang dapat dilihat dari frekuensi kemunculan di tiap dokumen (Fauzi, Arifin, & Yuniarti, 2014).

Pembobotan yang dilakukan dengan *TF.IDF* (Term Frequency-Inverse Document), merupakan metode yang digunakan dalam pembobotan hubungan kata atau *term* pada dokumen, dimana metode ini menggabungkan dua konsep perhitungan bobot, yaitu *Term Frequency* (TF) dan *Document Frequency* (DF) (Rosid, Gunawan, & Pramana, 2015). Berikut merupakan beberapa metode yang ada pada pembobotan (Fauzi, Arifin, & Yuniarti, 2014), diantaranya:

1. *Term Frequency* (TF)

Metode ini merupakan metode sederhana dalam pembobotan *term*, dimana tiap *term* diasumsi memiliki suatu kepentingan yang proposional terhadap kemunculan *term* pada dokumen. Bobot *term t* pada dokumen *d*, yaitu:

$$W_{tf}(t,d) = 1 + f(t,d) = \begin{cases} 1 + \log_{10} tf_{t,d}\ , & if\ tf_{t,d} > 0 \\ 0, & otherwise \end{cases} \quad (2.1)$$

$f(t,d) =$ frekuensi kemunculan pada *term t* dalam dokumen *d*

2. *Invers Document Frequency* (IDF)

TF lebih terfokus pada kemunculan *term* di dalam dokumen, sedangkan IDF lebih terfokus pada kemunculan *term* pada seluruh kumpulan dokumen. Pada pembobotan ini, *term* yang jarang muncul di kumpulan dokumen memiliki nilai kepentingan. Nilai kepentingan *term* diasumsi memiliki proporsi yang berkebalikan dengan jumlah dokumen dengan *term* di dalamnya.

$$idf_t = log_{10}(N/df_t), \quad (2.2)$$

N = jumlah seluruh dokumen

$df(t)$ = jumlah dokumen yang memiliki *term t.*

3. *TF.IDF*

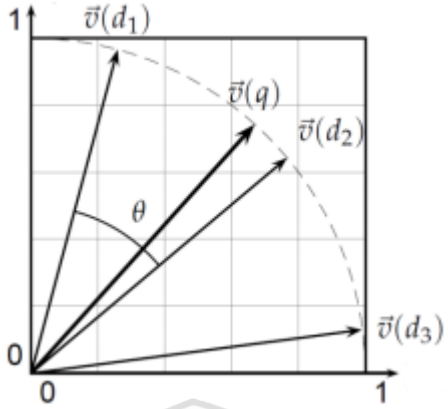
TF.IDF dihasilkan dari penggabungan perkalian dari rumus di atas, yaitu rumus TF dengan IDF sehingga kombinasi bobotnya, ialah:

$$W_{t,f}(t,d) \times idf_t \quad (2.3)$$

2.4.4 *Cosine Similarity*

Hasil dari pembobotan kata yang didapatkan nantinya akan dijadikan representasi vector, dimana representasi vector tersebut dapat dihitung dengan menggunakan *cosine similarity* atau nilai kemiripan pada suatu dokumen dengan *query* (Fauzi, Arifin, & Yuniarti, 2014). Konsep dari perhitungan ini dengan menghitung nilai cosine pada sudut antara dua vector, yaitu diberikan dokumen yang mewakili vector d_j dan *query q*, serta *term t* diekstrak dari database (Herlambang, Putri, & Wihandika, 2017). Berikut merupakan representasi

perumusan *cosine similarity* dalam bidang katersian yang ditunjukkan pada Gambar 2.2 (Fauzi, Arifin, & Yuniarti, 2014):



Gambar 2.2 Representasi *Cosine Similarity*

Pada gambar di atas, memiliki tiga vector, yaitu *d1, d2,* dan *d3*, serta satu vector *query q. Cosine similarity,* menghitug nilai dari kosinus θ dari *query* dengan tiga dokumen lainnya, dimana nilai ini menunjukkan derajat kemiripan dokumen yang ada dengan *query* (Fauzi, Arifin, & Yuniarti, 2014).

Untuk mendapatkan hasil yang lebih baik dan lebih terstruktur, maka akan lebih baik dilakukan normalisasi terlebih dahulu dengan rumus berikut:

$$W_{t,d} = \frac{W_{t,d}}{\sqrt{\sum_{t=1}^{n} {W_{t,d}}^2}} \quad (2.4)$$

Selanjutnya dilakukan perhitungan pada Cosine Similarity:

$$CosSim(d_j, q) = \overrightarrow{d_j}.\vec{q} = \sum_{i=1}^{t}(W_{ij}.W_{iq}) \quad (2.5)$$

2.4.5 *Query Expansion*

Query expansion, merupakan salah satu teknik dasar dari *Relevance Feedback,* dimana sistem nantinya akan menambah query tambahan pada pencarian kedua dari hasil pencarian pertama (Pamungkas, Indriati, & Ridok, 2015, mengutip dari Fachruddin, 2011). Model ini menggunakan perluasan query untuk meningkatkan kinerja pencarian (Saneifar, Bonniol, Poncelet, & Roche, 2014). Berikut merupakan teknik dari *query expansion* (Pamungkas, Indriati, & Ridok, 2015 mengutip dari Selberg, 1997):

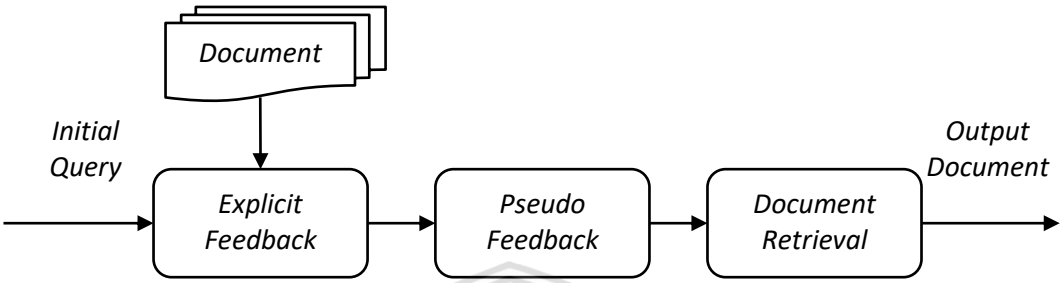
1. *Manual Query Expansion* (MQE)
   Pada teknik ini pengguna sendirilah yang melakukan modifikasi *query* sehingga sistem tidak membantu pengguna.

2. *Automatic Query Expansion* (AQE)
   Merupakan teknik yang memodifikasi *query* tanpa adanya control dari pengguna. Seperti misalnya pada sebuah sistem yang selalu menambahkan istilah atau sinonim dari query awal untuk *query* baru yang akan dianggap sistem AQE.

3. *Interaktive Query Expansion* (IQE)

Pada teknik ini membutuhkan interaksi antara pengguna dengan sistem dalam proses *query expansion.*

Berikut merupakan Diagram Tahapan *Query expansion* (Ludviani, Hayati, Arifin, & Purwitasari, 2015), yang ditunjukkan pada Gambar 2.3.

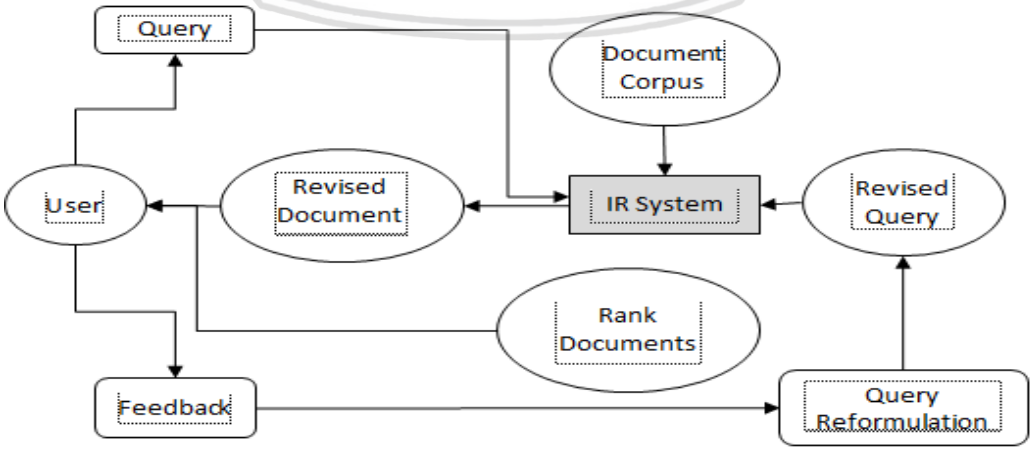


Gambar 2.3 Diagram Tahapan *Query Expansion*

2.4.6 *Relevance Feedback*

Pada metode ini mengacu pada proses interaktif, dimana membantu untuk meningkatkan kinerja pada proses *retrieval.* Dalam proses ini pengguna akan mengajukan suatu *query* yang kemudian sistem akan mengembalikan dokumen awal dan meminta kepada pengguna untuk melakukan penilaian terhadap dokumen, apakah dokumen tersebut relevan atau tidak. Kemudian sistem akan merumuskan ulang *query* berdasar penilaian pengguna (Saneifar, Bonniol, Poncelet, & Roche, 2014).

Relevance feedback, merupakan teknik yang ditemukan pertama kali oleh *Rocchio,* dimana teknik ini memodifikasi suatu *query* yang sering diimplementasikan pada *information retrieval*. Cara kerjanya ialah dengan memilih *term* penting dalam dokumen yang dianggap dokumen relevan oleh pengguna, serta menambahkan *term* penting ke dalam proses modifikasi *query* (Pamungkas, Indriati, & Ridok, 2015). Berikut ditunjukkan pada Gambar 2.4 arsitektur dari *Relevance Feedback* (Alam & Sadaf, 2015).



Gambar 2.4 Arsitektur *Relevance Feedback*

Berikut ini merupakan tiga metode *relevance feedback* (Pamungkas, Indriati, & Ridok, 2015), diantaranya:

1. *User Judgement (Explicit Relevance Feedback)*

   Metode ini terfokus pada penilaian relevan dari relevansi suatu dokumen dengan *query* tertentu, dimana penilai relevannya ialah penialaian yang ditafsirkan oleh pengguna.

2. *User Behavior (Implicit Relevance Feedback)*

Metode ini terfokus pada perilaku pengguna, seperti mencatat dokumen-dokumen yang terpilih maupun tidak, serta surasi waktu yang digunakan untuk melihat dokumen atau proses selama *browsing* hingga *scrolling* halaman.

3. *Top K Relevance Feedback (Blind/Pseudo Relevance Feedback)*

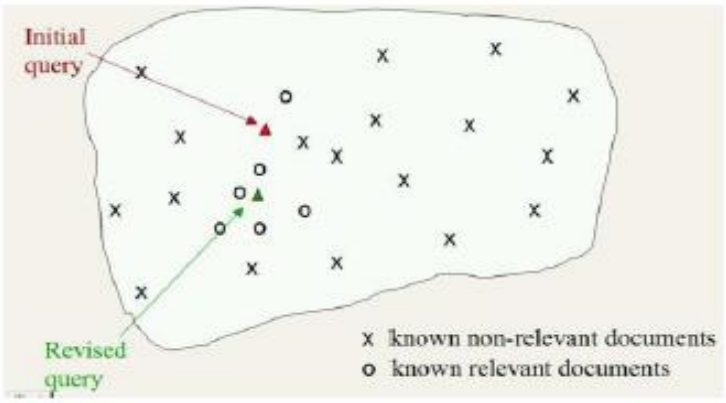
   Metode ini tidak melibatkan adanya pengguna dalam memperoleh keputusan, dimana cara kerjanya dengan mengambil dokumen teratas, serta secara sederhana mengasumsi dokumen mana yang relevan. *Pseudo Relevance Feedback,* dapat diterapkan ke dalam sistem *retrieval* tanpa memerlukan *feedback* dari pengguna*,* serta terbukti cocok untuk *enchacing retrieval* (Hazimeh & Zhai, 2015).

2.4.7 *User Judgement (Explicit Relevance Feedback)*

Relevance feedback jenis ini mendapatkan *feedback* secara *explicit* dari pengguna untuk menunjukkan penilaiannya. Hal ini terfokus dari bagaimana memperbaiki suatu struktur yang buruk ataupun *query* yang ambigu (Alam & Sadaf, 2015). Pengguna secara eksplisit memberikan penilaian pada dokumen, yaitu berupa dokumen relevan maupun tidak relevan dari perangkingan dokumen hasil dari *query* yang dimasukkan.

2.4.8 *Extended Rocchio Relevance Feedback*

Metode *Rocchio Relevance Feedback*, merupakan strategi reformulasi query yang popular, karena digunakan sebagai metode untuk membantu pengguna yang masih awam dengan *information retrieval* sistem (Yugianus, Dachlan, & Hasanah, 2013). Agoritma tersebut telah terbukti meningkatkan hasil pencarian (Jordan & Watters, 2004). Algortima *Rocchio* disebut dengan algortima klasik yang digunakan untuk mengimplementasikan *relevance feedback*. Pada konteks *query, information retrieval* memiliki query dari inputan pengguna dan akan menghasilkan dokumen yang relevan maupun tak relevan, seperti Gambar 2.5 berikut ini yang menunjukkan bentuk Algoritma *Rocchio* (Pamungkas, Indriati, & Ridok, 2015):



Gambar 2.5 Algoritme *Rocchio*

Proses pencarian yang dihasilkan berupa dokumen relevan maupun tidak relevan, dimana *query* yang diiputkan oleh pengguna berada pada *centroid* dari keseluruhan dokumen. Selanjutnya *query* baru akan berada pada *centroid* dokumen yang relevan. Berikut merupakan persamaan pada *query* modifikasi (Pamungkas, Indriati, & Ridok, 2015):

$$\overrightarrow{q_m} = \alpha . \overrightarrow{q_0} + \beta \frac{1}{|D_r|} . \sum_{\overrightarrow{d_j} \epsilon D_r} \overrightarrow{d_j} - \gamma \frac{1}{|D_n|} . \sum_{\overrightarrow{d_k} \epsilon D_n} \overrightarrow{d_k} \quad (2.6)$$

dimana:

q_m = *vector query* baru

q_0 = *vector query* awal

$\overrightarrow{d_j}$ = dokumen *vector* relevan

$\overrightarrow{d_k}$ = dokumen *vector* tidak relevan

D_r = dokumen relevan

D_n = dokumen tidak relevan

α = bobot *query* asli

β = bobot dokumen relevan

ϒ = bobot dokumen tidak relevan

Sedangkan Algortima *Extended Rocchio Relevance Feedback*, merupakan metode modifikasi dari metode *Rocchio Relevance Feedback*. Algoritme ini menunjukkan peningkatan kinerja pada saat pengambilan dokumen VSR atau *Vector Spave Retrieval*, serta akan mendapatkan hasil yang sebanding dengan algoritme tradisionalnya, yaitu *Rocchio Relevance Feedback*. *Term vector* yang dibangun dengan menggunakan proses TF/IDF. Nantinya tiap *vector* istilah akan merepresentasikan konteks tertentu (Jordan & Watters, 2004). Berikut merupakan 3 pendekatan yang dilakukan dalam proses *Extended Rocchio Relevance Feedback* (Jordan & Watters, 2004), diataranya:

1. *Query Modification*

Pada proses ini, saat pengguna menginputkan *query*, maka *query terms* akan dibandingkan nilai *term vector*. Proses similarity digunakan untuk membentuk sudut antara 2 vektor dengan memanfaatkan *cosine similarity.*

$$Sim(Q,V) = \frac{Q \bullet V}{|Q| \times |V|} \quad (2.7)$$

dimana Q, merupakan *query* dan *V*, merupakan *term vector.* Pada proses modifikasi *query* ini memanfaatkan dua kondisi, yaitu:

1. Jika *term vector V* tidak memiliki similarity, maka tidak perlu dilakukan modifikasi *query*
2. Jika *term vector V* memiliki nilai similarity yang lebih besar dari nilai *threshold* $(V > \sigma)$, maka dilakukan *query* modifikasi (Q_{mod}). Q_{mod} dibentuk dari kombinasi *query* dengan *term vector* tersebut.

Rata-rata bobot (*average weight*) yang digunakan untuk tiap term yang memiliki nilai *vector* dan *query* nilai selain nilai 0.

2. *Relevance Feedback*

Modifikasi pada suatu *query* dilakukan untuk menunjukkan dokumen retrieval. Dokumen yang telah diranking akan digunakan untuk memberntuk tiga *vector*, sebagai berikut:

1. *Term Vector* P : berisi nilai rata-rata *term weight,* bagi term pada dokumen-dokumen relevan, namun tidak berasal dari original *query*, Q.
2. *Term Vector* N : berisi nilai rata-rata *term weight,* bagi term pada dokumen-dokumen tidak relevan, namun tidak berasal dari original *query*, Q.
3. *Term Vector* F : berisi nilai dari *term vector,* V, namun tidak berasal dari P, N, dan Q.

3. Profile Modification

Term vector yang dibuat pada tahap ke-2, digunakan untuk memodifikasi nilai *query*. Pada proses ini dilakukan dua kondisi, yaitu:

1. Jika pada tahap 1 tidak dilakukan modifikasi *query*, maka menggunakan rumus *Rocchio Relevance Feedback* atau *Rocchio* tradisional. Dengan arti lain bahwa jika σ>*V*, maka menggunakan rumus:

$$V_{new} = \alpha * Q + \beta * P - \gamma * N \quad (2.8)$$

2. Jika pada tahap 1 dilakukan modifikasi *query*, atau σ<*V* maka dilakukan perhitungan sebagai berikut:

$$V = \alpha * Q_{mod} + \beta * P - \gamma * N + \Delta * F \quad (2.9)$$

Nilai $\alpha, \beta, \text{dan } \gamma$, merupakan nilai konstan yang ada pada algoritme *Rocchio*. Sedangkan Δ, merupakan nilai konstanta yang mengatur kerusakan pada *term*.

2.4.9 *Precision, Recall, F-Measure* dan Akurasi

Proses umum yang dilakukan untuk mengukur kualitas dari *retrieval* ialah dengan menggunakan *precision* dan *recall. Precision*, merupakan proporsi dari set yang diperoleh secara relevan. Sedangkan *recall,* merupakan proporsi dari semua

data relevan yang di koleksi termasuk dengan hasil yang diperoleh maupun dikembalikan. *F-measure,* sendiri merupakan pengukuran pada klasifikasi pencarian dokumen, serta peforma dari *query classification* (Pamungkas, Indriati, & Ridok, 2015).

Evaluasi peforma efektivitas pada sistem klasifikasi teks dengan menggunakan standar *confussion matrix* yang berisi informasi klasifikasi yang sebenarnya dan merupakan prediksi klasifikasi oleh sistem (Pamungkas, Indriati, & Ridok, 2015). Berikut merupakan tabel dari *confussion matrix* (Pamungkas, Indriati, & Ridok, 2015):

Tabel 2.1 *Confussion Matrix*

		Actual Class (expectation)	
		+	-
Predicted Class (Observation)	+	TP	FP
	-	FN	TN

Keterangan:
TP : *True Positive*, dimana menunjukkan perangkingan sistem merupakan dokumen yang sesuai dengan *query*.
FP : *False Prositive,* dimana menunjukkan dokumen dalam hasil perangkingan sistem tidak sesuai dengan *query*.
FN : *False Negative,* dimana menunjukkan dokumen tidak termasuk dalam perangkingan sistem dan harusnya sesuai dengan *query.*
TN : *True Negative,* dimana menunjukkan dokumen tidak termasuk perangkingan sistem dan memang seharusnya tidak sesuai *query.*

Berikut merupakan rumus perhitungan *precision, recall,* dan *F-Measure* (Pamungkas, Indriati, & Ridok, 2015):

$$precision = \frac{TP}{TP+FP} \quad (2.10)$$

$$recall = \frac{TP}{TP+FN} \quad (2.11)$$

Semakin tinggi nilai akurasi yang didapatkan, maka akan menunjukkan kesesuaian nilai dari prediksi pengujian pada *ground truth* atau nilai actual.

$$akurasi = \frac{TP+TN}{TP+FP+TN+FN} * 100\% \quad (2.12)$$

$$F1 = \frac{2 \times recall \times precision}{recall+precision} \quad (2.13)$$

F1 *measure*, ialah bobot *harmonic mean* yang ada pada *recall* dan *precision.*

BAB 3 METODOLOGI PENELITIAN

Pada bab ini membahas tentang metode, teknik, hingga proses yang dilakukan selama penelitian yang dilakukan. Dalam metodologi ini berisi beberapa subbab, yaitu tipe penelitian, serta strategi dan rancangan penelitian.

3.1 Tipe Penelitian

Pada penelitian ini memanfaatkan tipe penelitian nonimplementatif, dimana tipe ini melakukan analisis yang akan dikaji dan menghasilkan suatu hasil analisis ilmiah sebagai fokus utamanya. Hasil dari teknik atau metode ini dapat digunakan untuk menghasilkan suatu survey, eksperimen,studi kasus, dan lain sebagainya.

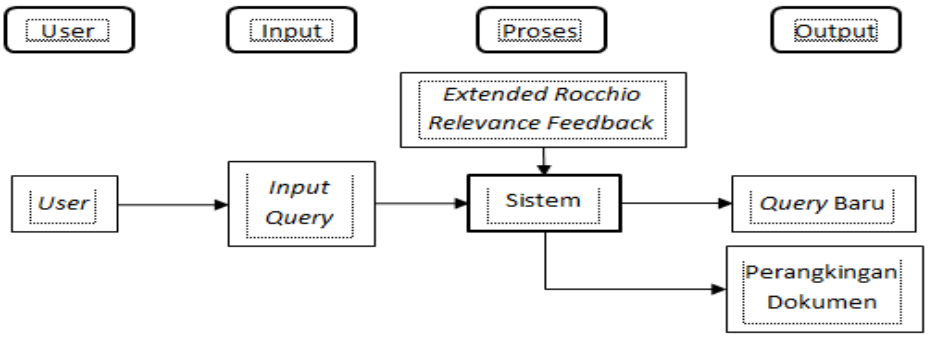
Jika ditinjau dalam jenis kegiatannya, pada penelitian ini memanfaatkan tipe penelitian analitik (*analytical/explanatory*). Tipe penelitian tersebut mengutamakan proses penggalian informasi yang bertujuan mengidentifikasi elemen penting dari objek penelitian yang digunakan sebagai dasar dalam pengambilan keputusan.

3.2 Strategi Penelitian

Penelitian ini menggunakan strategi penelitian eksperimen, dimana peneliian ini terfokus pada satu atau lebih variabel yang digunakan dengan cara tertentu sehingga mempengaruhi variabel terikat lainnya yang diukur untuk menguji hipotesis yang berhubungan dengan sebab akibat. Algoritma yang digunakan dalam eksperimen ini ialah dengan menggunakan metode pembobotan TF.IDF, *Cosine Similarity*, serta *Extended Rocchio Relevance Feedback*.

3.3 Rancangan Penelitian

Rancangan sistem ini digunakan untuk memberikan gambaran pada sistem mengenai bagaimana sistem berjalan, dimulai dari input, proses, hingga output. Berikut merupakan gambaran model perancangan arsitektur sistem yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Model Perancangan Arsitektur

Pengguna akan menginputkan suatu *query,* dimana nantinya sistem akan menampilkan *query* baru atau tambahan untuk mempermudah pengguna.

Kemudian pengguna akan mengkombinasikan *query* asli dengan *query-query* baru yang ditampilkan oleh sistem. Kemudian setelah memilih *query* yang akan diinputkan, sistem akan menampilkan dokumen-dokumen perangkingan hasil dari pencarian. Untuk menentukan dokumen relevan maupun tidak relevan, akan ditentukan dari *feedback* pengguna, yaitu dengan memberikan centang pada tiap dokumen.

3.3.1 Partisipan Penelitian

Pada penelitian ini ada beberapa pihak partisipan yang terlibat, yaitu penulis-penulis dalam berita yang tersedia dalam situs berita online LINE TODAY. Partisipan selanjutnya ialah tiga orang mahasiswa yang nantinya melakukan penilaian terhadap dokumen-dokumen yang ditampilkan dalam suatu *query* yang diinputkan. Penilaian berupa pemilihan dokumen relevan maupun tidak relevan hasil dari inputan *query*.

3.3.2 Lokasi Penelitian

Penelitian ini tidak memiliki lokasi khusus, melainkan dilakukan pada situs berita online LINE TODAY yang dapat diakses melalui web sehingga tidak memiliki lokasi kusus untuk penelitian yang dilakukan. Namun selama proses proses pembelajaran yang dilakukan berada di Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang.

3.3.3 Teknik Pengumpulan Data

Teknik pengumpulan data memanfaatkan teknik sekunder, dimana *dataset* yang digunakan berupa data yang telah tersedia dan merupakan dokumen yang ditulis dari laporan orang lain. Data yang didapatkan berasal dari situs berita online, yaitu LINE TODAY. Data berupa artikel-artikel berita yang ada di dalam situs berita tersebut. Nantinya berita-berita tersebut akan dilakukan proses *query expansion* dan dokumen relevan maupun tidak akan ditentukan oleh pengguna.

3.3.4 Teknik Pengujian

Pengujian dilakukan dengan menguji hasil kerja sistem yang dibuat dan dilakukan evaluasi sistem. Proses tersebut dilakukan guna mengetahui hasil sistem yang nantinya akan digunakan sebagai penarikan kesimpulan. Pengujian yang dilakukan dengan menilai dari hasil tingkat *akurasi*, *precision, recall*, dan *f-measure*. Selain itu, perbandingan juga dilakukan pada *query* sebelum dan sesudah dilakukan perhitungan dengan metode *Extended Rocchio Relevance Feedback*.

3.3.5 Peralatan Pendukung

Berikut merupakan spesifikasi kebutuhan sistem yang melibatkan perangkat keras dan perangkat lunak, diantaranya:

1. Perangkat keras yang dibutuhkan, ialah:
   - PC
   - RAM : 2 GB/ 4 GB/ 8GB

- ROM : 500 GB

2. Perangkat lunak yang digunakan untuk pengimplementasian pada sistem, ialah berbasis web sehingga membutuhkan:
   - Editor : Notepad++
   - Database : DBMS MySQL
   - Server : Xampp
   - Web Client : Chrome

3.4 Penarikan Kesimpulan dan Saran

Penarikan kesimpulan didapatkan setelah proses pengujian pada sistem selesai dikerjakan sehingga nantinya akan bisa diketahui efektifitas kinerja pada sistem. Selain itu, kesimpulan juga merupakan jawaban dari rumusan masalah yang dibahas sebelumnya.

Sedangkan pada saran digunakan untuk membantu pengembangan pada sistem selanjutnya, agar penelitian mengenai *query expansion* dapat tereksplor lebih baik lagi.

3.5 Jadwal Penelitian

Penelitian dilakukan dari bulan Februari hingga Juni yang memiliki jadwal penelitian sebagai berikut, ditunjukkan pada Tabel 3.1.

Tabel 3.1 Jadwal Penelitian

No	Uraian	Minggu ke-																			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
		Februari				Maret				April				Mei				Juni			
1	Konsultasi dan Penyusunan Laporan																				
2	Penyerahan Proposal																				
3	Perbaikan/Revisi Proposal Penelitian																				
4	Pengumpulan Data																				
5	Penyusunan Laporan Penelitian																				
6	Bimbingan dan Konsultasi Hasil Penelitian																				
7	Seminar Hasil Penelitian																				
8	Perbaikan Hasil Penelitian																				
9	Sidang Skripsi																				
10	Perbaikan Hasil Sidang Penelitian																				

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Pada bagian ini akan dibahas mengenai analisis dan perancangan pada sistem "*Query* E*xpansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback*".

4.1 Deskripsi Masalah

Dalam mendapatkan suatu informasi, dibutuhkan adanya artikel berita yang selalu *update* dan terkini. Kita tidak akan pernah kehabisan informasi, mengingat perkembangan zaman yang semakin pesat dan ditunjang dengan teknologi-teknologi canggih. Tidak hanya melalui media cetak, namun melalui media elektronik juga kini dapat dinikmati dengan mudah, guna mendapatkan informasi yang kita inginkan. Media elektronik cenderung lebih banyak digemari oleh masyarakat, karena mudah digunakan dimanapun dan kapanpun, karena dengan melalui *smartphone* yang terhubung dengan internet, kita dapat menikmati situs-situs berita *online* yang tentunya te-*update* dan terkini.

Untuk mengakses suatu berita yang diinginkan, kita membutuhkan adanya mesin telusur web atau *search engine*, guna mempersingkat waktu kita untuk menemukan berita apa saja yang kita inginkan untuk dibaca, mengingat dalam situs berita *online* biasanya terdiri dari puluhan berita yang di-*update* tiap harinya. Tak hanya sampai disitu, saat kita memasukkan *query* pada mesin pencarian, permasalahan yang sering dihadapi ialah *query* yang kita masukkan kurang spesifik sehingga berita yang ditampilkan terkadang tidak sesuai dengan yang kita inginkan. Dengan adanya ekspansi *query* tersebut, sangat membantu kita untuk mendapatkan berita-berita spesifik, karena akan diberikan rekomendasi *query-query* tambahan yang relevan dengan *query* yang kita masukkan sebelumnya.

4.2 Deskripsi Umum Sistem

Query Expansion Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback,* memanfaatkan sistem yang dibangun untuk mendapatkan query baru agar query lebih spesifik. *Query* bantuan atau tambahan yang diberikan sistem akan membantu pengguna dalam memberikan inputan *query* yang lebih spesifik sehingga dokumen yang didapatkan juga sesuai dengan yang diharapkan. Selain itu dengan adanya *user judgement*, pencarian dokumen dari *query* yang diberikan akan lebih akurat, karena berdasar penilaian dari pengguna atau *user*. Data latih yang didapatkan akan dilakukan *preprocessing, term weighting, normalisasi, cosine similarity,* hingga dengan metode *extended rocchio relevance feedback* untuk menentukan dokumen yang relevan serta *query* baru.

4.3 Manualisasi

Pada bagian ini dilakukan proses manualisasi dengan memanfaatkan beberapa tahapan, yaitu *preprocessing, TF-IDF, Cosine Similarity,* hingga metode *Extended Rocchio Relevance Feedback*

4.3.1 *Preprocessing*

Pada tahapan ini digunakan untuk mendapatkan data yang lebih terstruktur, guna mempermudah proses perhitungan pada tahapan selanjutnya. Tahapan *preprocessing* terdiri dari *cleansing, case folding, tokenization, filtering,* dan *stemming*. Berikut merupakan data latih yang terdiri dari beberapa kategori berita di LINE TODAY, yaitu 2 berita dari kategori *News,* 2 berita dari kategori *Sci-Tech,* dan 1 berita dari kategori *Showbiz*, yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Data Latih

	Isi Dokumen
Dok 1	BRI soal Misteri Raibnya Uang Nasabah: Begitu ada Pengaduan Dijamin Diganti Komisaris Utama Bank Rakyat Indonesia (BRI), Andrinof Chaniago, memastikan dana nasabah yang hilang akibat kejahatan teknologi perbankan akan diganti. Dana penggantian akan didapat tak lama usai nasabah melaporkan tindak kejahatan ini. "Itu cepat, begitu ada pengaduan dijamin diganti," ujarnya saat ditemui di Istana Negara, Jakarta, Kamis (15/3). Pihaknya menduga tindak kejahatan ini dilakukan oleh jaringan luar negeri. Maka dari itu, jika kejadiannya pembobolan data nasabah, maka nasabah yang menjadi korban akan mendapat ganti dana.
Dok 2	Kasus Dana Nasabah BRI Raib Kartu Debit Non Chip Lebih Beresiko Anggota Badan Pengawas Asosiasi Sistem Pembayaran Indonesia (ASPI) Kresno Sediarsi turut mengomentari kasus raibnya dana nasabah Bank Rakyat Indonesia atau BRI Unit Ngadiluwih, di Kediri, Jawa Timur. Senada dengan pihak kepolisian sebelumnya, Kresno menduga dana nasabah yang hilang itu diduga karena adanya praktik skimming.
Dok 3	Kata Sandiaga, PSO untuk Dharma Jaya Tak Kunjung Cair karena Hal Ini Wakil Gubernur DKI Jakarta Sandiaga Uno mengatakan, dana *public service obligation* (PSO) untuk PD Dharma Jaya segera cair. Hal ini dia ketahui setelah mendapat laporan dari Kepala Badan Pengelola Keuangan Daerah (BPKD) DKI Jakarta Michael Rolandi. “Kalau dari Pak Michael seharusnya sudah keluar ya, satu dua hari katanya,” ujar Sandiaga di Balai Kota DKI, Kamis (15/3/2018). Oleh karena itu, Sandiaga meminta Direktur Utama PD Dharma Jaya Marina Ratna Dwi Kusumajati bersabar. Menurut Sandiaga, lambatnya pencairan dana tersebut karena permasalahan administrasi.
Dok 4	Jokowi: Pertumbuhan Ekonomi Butuh Bank yang Agresif Presiden menilai, capaian itu merupakan gambaran perbankan kurang berani mengambil risiko. “Kalau saya diberi angka itu, saya ambil 12% nya. Kembali lagi, risiko paling besar apabila kita tidak berani ambil risiko. Perbankan harus *prudent* dan hati-hati, iya saya setuju,” tandasnya. Dalam pertemuan itu, Presiden didampingi Ketua Dewan Komisioner OJK Wimboh Santoso, Menko Perekonomian Darmin Nasution, dan Menteri Keuangan Sri Mulyani Indrawati. Dari kalangan perbankan, Direktur Utama 4 bank BUMN hadir, demikian juga bank-bank swasta nasional lainnya.

Tabel 4.1 Data Latih (lanjutan)

	Isi Dokumen
Dok 5	Sejumlah SMA di Sulsel Laksanakan USBN Menggunakan Android Siswa di sejumlah Sekolah Menengah Atas (SMA) di Provinsi Sulawesi Selatan (Sulsel) mengikuti Ujian Sekolah Berbasis Nasional (USBN) dengan menggunakan sistem operasi Android. "Kami mendorong agar bukan hanya ujian nasional, tetapi ujian sekolah juga sudah berbasis komputer, bahkan android," kata Kepala Dinas Pendidikan Sulsel Irman Yasin Limpo di Makassar, Senin (19/3/2018).
Query (Data Uji)	Dana Nasabah Bank

4.3.1.1 *Cleansing* dan *Case Folding*

Pada proses ini dilakukan dua tahap, yaitu penghapusan pada tag url, tanda baca, angka, atau karakter lainnya selain huruf dan merubah semua kata menjadi huruf kecil atau *lower case*. Penghapusan pada karakter-karakter tersebut digunakan untuk mengurangi *noise* yang ada pada data. Berikut merupakan hasil *cleansing* dan *case folding* yang dilakukan pada data uji, yang ditunjukkan pada Tabel 4.2.

Tabel 4.2 Hasil *Cleansing* dan *Case Folding*

	Isi Dokumen
Dok 1	bri soal misteri raibnya uang nasabah begitu ada pengaduan dijamin diganti komisaris utama bank rakyat indonesia bri andrinof chaniago memastikan dana nasabah yang hilang akibat kejahatan teknologi perbankan akan diganti dana penggantian akan didapat tak lama usai nasabah melaporkan tindak kejahatan ini itu cepat begitu ada pengaduan dijamin diganti ujarnya saat ditemui di istana negara jakarta kamis pihaknya menduga tindak kejahatan ini dilakukan oleh jaringan luar negeri maka dari itu jika kejadiannya pembobolan data nasabah maka nasabah yang menjadi korban akan mendapat ganti dana
Dok 2	kasus dana nasabah bri raib kartu debit non chip lebih beresiko anggota badan pengawas asosiasi sistem pembayaran indonesia aspi kresno sediarsi turut mengomentari kasus raibnya dana nasabah bank rakyat indonesia atau bri unit ngadiluwih di kediri jawa timur senada dengan pihak kepolisian sebelumnya kresno menduga dana nasabah yang hilang itu diduga karena adanya praktik skimming
Dok 3	kata sandiaga pso untuk dharma jaya tak kunjung cair karena hal ini wakil gubernur dki jakarta sandiaga uno mengatakan dana *public service obligation* pso untuk pd dharma jaya segera cair hal ini dia ketahui setelah mendapat laporan dari kepala badan pengelola keuangan daerah bpkd dki jakarta michael rolandi kalau dari pak michael seharusnya sudah keluar ya satu dua hari katanya ujar sandiaga di balai kota dki kamis oleh karena itu sandiaga meminta direktur utama pd dharma jaya marina ratna dwi kusumajati bersabar menurut sandiaga lambatnya pencairan dana tersebut karena permasalahan administrasi

Tabel 4.2 Hasil *Cleansing* dan *Case Folding* (lanjutan)

	Isi Dokumen
Dok 4	jokowi pertumbuhan ekonomi butuh bank yang agresif presiden menilai capaian itu merupakan gambaran perbankan kurang berani mengambil risiko kalau saya diberi angka itu saya ambil nya kembali lagi risiko paling besar apabila kita tidak berani ambil risiko perbankan harus *prudent* dan hati hati iya saya setuju tandasnya dalam pertemuan itu presiden didampingi ketua dewan komisioner ojk wimboh santoso menko perekonomian darmin nasution dan menteri keuangan sri mulyani indrawati dari kalangan perbankan direktur utama bank bumn hadir demikian juga bank bank swasta nasional lainnya
Dok 5	sejumlah sma di sulsel laksanakan usbn menggunakan android siswa di sejumlah sekolah menengah atas sma di provinsi sulawesi selatan sulsel mengikuti ujian sekolah berbasis nasional usbn dengan menggunakan sistem operasi android kami mendorong agar bukan hanya ujian nasional tetapi ujian sekolah juga sudah berbasis komputer bahkan android kata kepala dinas pendidikan sulsel irman yasin limpo di makassar senin
Query (Data Uji)	dana nasabah bank

4.3.1.2 *Tokenization*

Pada proses ini dilakukan dengan memisahkan tiap kata dengan *whitespace* dari data dokumen. Berikut merupakan Tabel 4.3 yang menunjukkan hasil dari tokenisasi setelah dilakukan proses sebelumnya, yaitu *cleansing* dan *case folding*

Tabel 4.3 Hasil *Tokenization*

	Isi Dokumen
Dok 1	bri soal misteri raibnya uang nasabah begitu ada pengaduan dijamin diganti komisaris utama bank rakyat indonesia bri andrinof chaniago memastikan dana nasabah yang hilang akibat kejahatan teknologi perbankan akan diganti dana penggantian akan didapat tak lama usai nasabah melaporkan tindak kejahatan ini itu cepat begitu ada pengaduan dijamin diganti ujarnya saat ditemui di istana negara jakarta kamis pihaknya menduga tindak kejahatan ini dilakukan oleh jaringan luar negeri maka dari itu jika kejadiannya pembobolan data nasabah maka nasabah yang menjadi korban akan mendapat ganti dana
Dok 2	kasus dana nasabah bri raib kartu debit non chip lebih beresiko anggota badan pengawas asosiasi sistem pembayaran indonesia aspi kresno sediarsi turut mengomentari kasus raibnya dana nasabah bank rakyat indonesia atau bri unit ngadiluwih di kediri jawa timur senada dengan pihak kepolisian sebelumnya kresno menduga dana nasabah yang hilang itu diduga karena adanya praktik skimming
Dok 3	kata sandiaga pso untuk dharma jaya tak kunjung cair karena hal ini wakil gubernur dki jakarta sandiaga uno mengatakan dana *public service obligation* pso untuk pd dharma jaya segera cair hal ini dia ketahui setelah mendapat laporan dari kepala badan pengelola keuangan daerah

Tabel 4.3 Hasil *Tokenization* (lanjutan)

	Isi Dokumen
Dok 3	bpkd dki jakarta michael rolandi kalau dari pak michael seharusnya sudah keluar ya satu dua hari katanya ujar sandiaga di balai kota dki kamis oleh karena itu sandiaga meminta direktur utama pd dharma jaya marina ratna dwi kusumajati bersabar menurut sandiaga lambatnya pencairan dana tersebut karena permasalahan administrasi
Dok 4	jokowi pertumbuhan ekonomi butuh bank yang agresif presiden menilai capaian itu merupakan gambaran perbankan kurang berani mengambil risiko kalau saya diberi angka itu saya ambil nya kembali lagi risiko paling besar apabila kita tidak berani ambil risiko perbankan harus *prudent* dan hati hati iya saya setuju tandasnya dalam pertemuan itu presiden didampingi ketua dewan komisioner ojk wimboh santoso menko perekonomian darmin nasution dan menteri keuangan sri mulyani indrawati dari kalangan perbankan direktur utama bank bumn hadir demikian juga bank bank swasta nasional lainnya
Dok 5	sejumlah sma di sulsel laksanakan usbn menggunakan android siswa di sejumlah sekolah menengah atas sma di provinsi sulawesi selatan sulsel mengikuti ujian sekolah berbasis nasional usbn dengan menggunakan sistem operasi android kami mendorong agar bukan hanya ujian nasional tetapi ujian sekolah juga sudah berbasis komputer bahkan android kata kepala dinas pendidikan sulsel irman yasin limpo di makassar senin
Query (Data Uji)	dana nasabah bank

4.3.1.3 *Filtering*

Filtering ini dilakukan dengan melakukan penghapusan pada kata yang deskriptif atau kurang penting. Kata-kata kurang penting yang berada dalam list *stopword*, dimana jika kata yang tersedia dalam list tersebut akan dihilangkan. Beberapa contoh kata dalam *stopword*, seperti aku, di, ke, ujarnya, dan lain sebagainya. Pada Tabel 4.4 ini akan menunjukkan hasil dari *filtering* yang telah dilakukan pada data latih setelah proses sebelumnya, yaitu tokenisasi.

Tabel 4.4 Hasil *Filtering*

	Isi Dokumen
Dok 1	bri misteri raibnya uang nasabah pengaduan dijamin diganti komisaris utama bank rakyat indonesia bri andrinof chaniago dana nasabah hilang akibat kejahatan teknologi perbankan diganti dana penggantian nasabah melaporkan tindak kejahatan cepat pengaduan dijamin diganti ditemui istana negara jakarta kamis menduga tindak kejahatan jaringan negeri kejadiannya pembobolan data nasabah nasabah korban ganti dana
Dok 2	dana nasabah bri raib kartu debit non chip beresiko anggota badan pengawas asosiasi sistem pembayaran indonesia aspi kresno sediarsi mengomentari raibnya dana nasabah bank rakyat indonesia bri unit ngadiluwih kediri jawa timur senada kepolisian kresno

Tabel 4.4 Hasil *Filtering* (lanjutan)

	Isi Dokumen
Dok 2	menduga dana nasabah hilang diduga praktik skimming
Dok 3	sandiaga pso dharma jaya kunjung cair wakil gubernur dki jakarta sandiaga uno dana *public service obligation* pso pd dharma jaya cair ketahui laporan kepala badan pengelola keuangan daerah bpkd dki jakarta michael rolandi pak michael sandiaga balai kota dki kamis sandiaga direktur utama pd dharma jaya marina ratna dwi kusumajati bersabar sandiaga lambatnya pencairan dana permasalahan administrasi
Dok 4	jokowi pertumbuhan ekonomi butuh bank agresif presiden menilai capaian gambaran perbankan berani mengambil risiko angka ambil risiko berani ambil risiko perbankan *prudent* hati hati setuju pertemuan presiden didampingi ketua dewan komisioner ojk wimboh santoso menko perekonomian darmin nasution menteri keuangan sri mulyani indrawati kalangan perbankan direktur utama bank bumn hadir bank bank swasta nasional
Dok 5	sma sulsel laksanakan usbn android siswa sekolah menengah atas sma provinsi sulawesi selatan sulsel mengikuti ujian sekolah berbasis nasional usbn sistem operasi android mendorong ujian nasional ujian sekolah berbasis komputer android kepala dinas pendidikan sulsel irman yasin limpo makassar senin
Query (Data Uji)	dana nasabah bank

4.3.1.4 *Stemming*

Proses *stemming* ini dilakukan dengan mengubah kata menjadi kata dasar. Pada proses ini dilakukan dengan berdasarkan algoritme Nazief-Adriani. Berikut merupakan Tabel 4.5 merupakan hasil proses *stemming* yang dilakukan dari data uji pada proses sebelumnya, yaitu proses *filtering*.

Tabel 4.5 Hasil *Stemming*

	Isi Dokumen
Dok 1	bri misteri raib uang nasabah adu jamin ganti komisaris utama bank rakyat indonesia bri andrinof chaniago dana nasabah hilang akibat jahat teknologi bank ganti dana ganti nasabah lapor tindak jahat cepat adu jamin ganti temu istana negara jakarta kamis duga tindak jahat jaring negeri jadi bobol data nasabah nasabah korban ganti dana
Dok 2	dana nasabah bri raib kartu debit non chip resiko anggota badan pengawas asosiasi sistem bayar indonesia aspi kresno sediarsi komentar raib dana nasabah bank rakyat indonesia bri unit ngadiluwih kediri jawa timur nada kepolisian kresno duga dana nasabah hilang duga praktik skimming
Dok 3	sandiaga pso dharma jaya kunjung cair wakil gubernur dki jakarta sandiaga uno dana *public service obligation* pso pd dharma jaya cair tahu lapor kepala badan pengelola keuangan daerah bpkd dki jakarta michael rolandi pak michael sandiaga

Tabel 4.5 Hasil *Stemming* (lanjutan)

	Isi Dokumen
Dok 3	balai kota dki kamis sandiaga direktur utama pd dharma jaya marina ratna dwi kusumajati sabar sandiaga lambat cair dana masalah administrasi
Dok 4	jokowi tumbuh ekonomi butuh bank agresif presiden nilai capai gambar bank berani ambil risiko angka ambil risiko berani ambil risiko bank *prudent* hati hati tuju temu presiden didampingi ketua dewan komisioner ojk wimboh santoso menko ekonomi darmin nasution menteri keuangan sri mulyani indrawati kalang bank direktur utama bank bumn hadir bank bank swasta nasional
Dok 5	sma sulsel laksana usbn android siswa sekolah menengah atas sma provinsi sulawesi selatan sulsel ikut ujian sekolah berbasis nasional usbn sistem operasi android dorong uji nasional uji sekolah basis komputer android kepala dinas pendidikan sulsel irman yasin limpo makassar senin
Query (Data Uji)	dana nasabah bank

4.3.2 Pembobotan TF.IDF

Pada proses pembobotan ini dilakukan dengan menggunakan TF.IDF untuk memberikan nilai bobot pada tiap *term* dan mengukur keunikan kata. Berikut merupakan proses yang dilakukan pada pembobotan TF.IDF setelah proses *preprocessing* selesai, dimana hasil paling akhir ialah hasil dari proses *stemming* yang kemudian dipilih kembali *term* uniknya, karena untuk memperjelas proses perhitungan agar lebih singkat dan mudah dipahami, yang ditunjukkan pada Tabel 4.6 dan Tabel 4.7.

Tabel 4.6 *Term* Unik Data Latih Hasil *Preprocessing*

Dok 1	Dok 2	Dok 3	Dok 4	Dok 5
bank	aspi	administrasi	agresif	android
bobol	atm	dana	bank	komputer
bri	bank	pso	bumn	operasi
dana	bri		ekonomi	pendidikan
data	chip		nasional	sekolah
hilang	dana		presiden	sistem
jamin	debit		risiko	sma
korban	hilang			uji
lapor	kartu			ujian
misteri	kediri			usbn
nasabah	nasabah			
negara	praktik			
raib	raib			
teknologi	sistem			
transaksi	skimming			

Pada Tabel 4.7 dibawah ini merupakan data uji manualisasi yang dianggap sebagai *query* asli.

Tabel 4.7 Data Uji Hasil *Preprocessing*

Query (Data Uji)
bank
dana
nasabah

4.3.2.1 *Term Frequency* (TF)

Term-term unik yang dihasilkan di tiap dokumen, kemudian dilakukan perhitungan banyak kemunculan atau *frequency term* atau kata yang muncul di tiap dokumen maupun *query.* Berikut merupakan *term frequency* kemunculan *term* unik di tiap dokumen yang ditunjukkan pada Tabel 4.8.

Tabel 4.8 *Term Frequency*

No	Term	TF					
		Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query
1	administrasi	0	0	1	0	0	0
2	agresif	0	0	0	1	0	0
3	android	0	0	0	0	3	0
4	aspi	0	2	0	0	0	0
5	atm	0	4	0	0	0	0
6	bank	2	1	0	7	0	1
7	bobol	1	0	0	0	0	0
8	bri	2	2	0	0	0	0
9	bumn	0	0	0	1	0	0
10	chip	0	1	0	0	0	0
11	dana	3	3	2	0	0	1
12	data	2	0	0	0	0	0
13	debit	0	1	0	0	0	0
14	ekonomi	0	0	0	1	0	0
15	hilang	1	1	0	0	0	0
16	jamin	2	0	0	0	0	0
17	kartu	0	1	0	0	0	0
18	komputer	0	0	0	0	1	0
19	korban	1	0	0	0	0	0
20	lapor	1	0	0	0	0	0
21	misteri	1	0	0	0	0	0
22	nasabah	4	3	0	0	0	1
23	nasional	0	0	0	1	0	0
24	negara	1	0	0	0	0	0

Tabel 4.8 *Term Frequency* (lanjutan)

No	Term	TF					
		Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query
25	operasi	0	0	0	0	1	0
26	pendidikan	0	0	0	0	1	0
27	praktik	0	1	0	0	0	0
28	presiden	0	0	0	2	0	0
29	pso	0	0	2	0	0	0
30	raib	1	2	0	0	0	0
31	risiko	0	0	0	3	0	0
32	sekolah	0	0	0	0	3	0
33	sistem	0	1	0	0	1	0
34	skimming	0	4	0	0	0	0
35	sma	0	0	0	0	2	0
36	teknologi	1	0	0	0	0	0
37	transaksi	3	0	0	0	0	0
38	uji	0	0	0	0	2	0
39	ujian	0	0	0	0	1	0
40	usbn	0	0	0	0	2	0

4.3.2.2 TF *Weight* dan IDF

Proses perhitungan pada pembobotan ini dilakukan setelah hasil dari sebelumnya, yaitu mencari *frequency* kata atau *term* di tiap dokumen. Pembobotan TF ini dilakukan dengan menggunakan rumus pada persamaan (2.1), seperti contoh berikut dilakukan perhitungan pada *term* atm di dokumen 2:

$$W_{tf}(t,d) = 1 + f(t,d) = \begin{cases} 1 + \log_{10} tf_{t,d}\ , & if\ tf_{t,d} > 0 \\ 0, & otherwise \end{cases}$$

$$= 1 + \log_{10} 4 = 1 + 0.60206 = 1.60206$$

Kemudian dilanjutkan dengan menghitung banyak dokumen *d* yang memiliki kata atau *term* t didalamnya (df). Setelah nilai df diketahui, kemudian dilakukan dengan menghitung IDF, dengan menggunakan persamaan (2.2), seperti contoh berikut dilakukan perhitungan pada *term* atm di dokumen 2:

$$idf_t = \log_{10}\left(\frac{N}{df_t}\right) = \log_{10}\frac{5}{1} = 0.69897$$

N menunjukkan banyak dokumen yang digunakan.

Berikut merupakan tabel hasil dari perhitungan TF dan IDF yang ditunjukkan pada Tabel 4.9.

Tabel 4.9 Hasil Perhitungan TF *Weight* dan IDF

No	Term	Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query	dft	idft
1	administrasi	0	0	1	0	0	0	1	0.69897
2	agresif	0	0	0	1	0	0	1	0.69897
3	android	0	0	0	0	1.47712	0	1	0.69897
4	aspi	0	1.30103	0	0	0	0	1	0.69897
5	atm	0	1.60206	0	0	0	0	1	0.69897
6	bank	1.30103	1	0	1.845098	0	1	3	0.22185
7	bobol	1	0	0	0	0	0	1	0.69897
8	bri	1.30103	1.30103	0	0	0	0	2	0.39794
9	bumn	0	0	0	1	0	0	1	0.69897
10	chip	0	1	0	0	0	0	1	0.69897
11	dana	1.47712	1.47712	1.30103	0	0	1	3	0.22185
12	data	1.30103	0	0	0	0	0	1	0.69897
13	debit	0	1	0	0	0	0	1	0.69897
14	ekonomi	0	0	0	1	0	0	1	0.69897
15	hilang	1	1	0	0	0	0	2	0.39794
16	jamin	1.30103	0	0	0	0	0	1	0.69897
17	kartu	0	1	0	0	0	0	1	0.69897
18	komputer	0	0	0	0	1	0	1	0.69897
19	korban	1	0	0	0	0	0	1	0.69897
20	lapor	1	0	0	0	0	0	1	0.69897
21	misteri	1	0	0	0	0	0	1	0.69897
22	nasabah	1.60206	1.47712	0	0	0	1	2	0.39794
23	nasional	0	0	0	1	0	0	1	0.69897
24	negara	1	0	0	0	0	0	1	0.69897
25	operasi	0	0	0	0	1	0	1	0.69897
26	pendidikan	0	0	0	0	1	0	1	0.69897
27	praktik	0	1	0	0	0	0	1	0.69897
28	presiden	0	0	0	1.30103	0	0	1	0.69897
29	pso	0	0	1.30103	0	0	0	1	0.69897
30	raib	1	1.30103	0	0	0	0	2	0.39794
31	risiko	0	0	0	1.47712	0	0	1	0.69897
32	sekolah	0	0	0	0	1.47712	0	1	0.69897
33	sistem	0	1	0	0	1	0	2	0.39794
34	skimming	0	1.60206	0	0	0	0	1	0.69897
35	sma	0	0	0	0	1.30103	0	1	0.69897
36	teknologi	1	0	0	0	0	0	1	0.69897
37	transaksi	1.47712	0	0	0	0	0	1	0.69897
38	uji	0	0	0	0	1.30103	0	1	0.69897

Tabel 4.9 Hasil Perhitungan TF *Weight* dan IDF (lanjutan)

No	Term	Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query	dft	idft
39	ujian	0	0	0	0	1	0	1	0.69897
40	usbn	0	0	0	0	1.30103	0	1	0.69897

4.3.2.3 *TF.IDF*

Perhitungan TF.IDF ini merupakan hasil perkalian antara TF dengan IDF yang sesuai dengan persamaan (2.3). Berikut merupakan contoh perhitungan dilakukan pada *term* atm di dokumen 2:

$$W_{t,d} = W_{tf}(t,d) * idf_t = 1.60206 * 0.69897 = 1.11979$$

Pada Tabel 4.10 menunjukkan hasil dari perhitungan TF.IDF.

Tabel 4.10 Hasil Perhitungan TF.IDF

No	Wtd	Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query
1	administrasi	0	0	0.69897	0	0	0
2	agresif	0	0	0	0.69897	0	0
3	android	0	0	0	0	1.03246	0
4	aspi	0	0.90938	0	0	0	0
5	atm	0	1.11979	0	0	0	0
6	bank	0.28863	0.22185	0	0.40933	0	0.22185
7	bobol	0.69897	0	0	0	0	0
8	bri	0.51773	0.51773	0	0	0	0
9	bumn	0	0	0	0.69897	0	0
10	chip	0	0.69897	0	0	0	0
11	dana	0.32770	0.32770	0.28863	0	0	0.22185
12	data	0.90938	0	0	0	0	0
13	debit	0	0.69897	0	0	0	0
14	ekonomi	0	0	0	0.69897	0	0
15	hilang	0.39794	0.39794	0	0	0	0
16	jamin	0.90938	0	0	0	0	0
17	kartu	0	0.69897	0	0	0	0
18	komputer	0	0	0	0	0.69897	0
19	korban	0.69897	0	0	0	0	0
20	lapor	0.69897	0	0	0	0	0
21	misteri	0.69897	0	0	0	0	0
22	nasabah	0.63752	0.58781	0	0	0	0.39794
23	nasional	0	0	0	0.69897	0	0
24	negara	0.69897	0	0	0	0	0
25	operasi	0	0	0	0	0.69897	0
26	pendidikan	0	0	0	0	0.69897	0
27	praktik	0	0.69897	0	0	0	0

Tabel 4.10 Hasil Perhitungan TF.IDF (lanjutan)

No	Wtd	Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query
28	presiden	0	0	0	0.90938	0	0
29	pso	0	0	0.90938	0	0	0
30	raib	0.39794	0.51773	0	0	0	0
31	risiko	0	0	0	1.03246	0	0
32	sekolah	0	0	0	0	1.03246	0
33	sistem	0	0.39794	0	0	0.39794	0
34	skimming	0	1.11979	0	0	0	0
35	sma	0	0	0	0	0.90938	0
36	teknologi	0.69897	0	0	0	0	0
37	transaksi	1.03246	0	0	0	0	0
38	uji	0	0	0	0	0.90938	0
39	ujian	0	0	0	0	0.69897	0
40	usbn	0	0	0	0	0.90938	0

4.3.2.4 Normalisasi TF.IDF

Pada proses ini dilakukan proses perhitungan normalisasi dari hasil perhitungan tabel sebelumnya, yaitu TF.IDF. Normalisasi dilakukan agar hasil lebih terstruktur dan dilakukan sesuai dengan persamaan (2.4). Berikut merupakan contoh perhitungan pada *term* atm di dokumen 2:

$$W_{t,d} = \frac{W_{t,d}}{\sqrt{\sum_{t=1}^{n} {W_{t,d}}^2}} = \frac{1.11979188}{\sqrt{2.57759593}} = 0.43443$$

Berikut ditunjukkan hasil perhitungan dari Normalisasi pada Tabel 4.11.

Tabel 4.11 Hasil Normalisasi TF.IDF

No	Wtd	Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query
1	administrasi	0	0	0.59098	0	0	0
2	agresif	0	0	0	0.34884	0	0
3	android	0	0	0	0	0.39812	0
4	aspi	0	0.35280	0	0	0	0
5	atm	0	0.43443	0	0	0	0
6	bank	0.11042	0.08607	0	0.20429	0	0.43779
7	bobol	0.26739	0	0	0	0	0
8	bri	0.19806	0.20086	0	0	0	0
9	bumn	0	0	0	0.34884	0	0
10	chip	0	0.27117	0	0	0	0
11	dana	0.12536	0.12713	0.24404	0	0	0.43779
12	data	0.34788	0	0	0	0	0
13	debit	0	0.27117	0	0	0	0

Tabel 4.11 Hasil Normalisasi TF.IDF (lanjutan)

No	Wtd	Dok 1	Dok 2	Dok 3	Dok 4	Dok 5	Query
14	ekonomi	0	0	0	0.34884	0	0
15	hilang	0.15223	0.15438	0	0	0	0
16	jamin	0.34788	0	0	0	0	0
17	kartu	0	0.27117	0	0	0	0
18	komputer	0	0	0	0	0.26952	0
19	korban	0.26739	0	0	0	0	0
20	lapor	0.26739	0	0	0	0	0
21	misteri	0.26739	0	0	0	0	0
22	nasabah	0.24389	0.22804	0	0	0	0.78529
23	nasional	0	0	0	0.34884	0	0
24	negara	0.26739	0	0	0	0	0
25	operasi	0	0	0	0	0.26952	0
26	pendidikan	0	0	0	0	0.26952	0
27	praktik	0	0.27117	0	0	0	0
28	presiden	0	0	0	0.45385	0	0
29	pso	0	0	0.76889	0	0	0
30	raib	0.15223	0.20086	0	0	0	0
31	risiko	0	0	0	0.51528	0	0
32	sekolah	0	0	0	0	0.39812	0
33	sistem	0	0.15438	0	0	0.15345	0
34	skimming	0	0.43443	0	0	0	0
35	sma	0	0	0	0	0.35066	0
36	teknologi	0.26739	0	0	0	0	0
37	transaksi	0.39497	0	0	0	0	0
38	uji	0	0	0	0	0.35066	0
39	ujian	0	0	0	0	0.26952	0
40	usbn	0	0	0	0	0.35066	0

4.3.3 Cosine Similarity

Proses ini dilakukan setelah proses normalisasi telah selesai dilakukan pada proses sebelumnya. Sesuai dengan persamaan (2.5) proses *cosine similarity* dilakukan, guna mengukur nilai kemiripan pada data latih terhadap data uji. Berikut merupakan contoh perhitungan pada *term* di dokumen 2:

$$CosSim(d_j, q) = \sum_{i=1}^{t} (W_{ij}.W_{iq})$$

$$= \begin{pmatrix} (0.3528 * 0) + (0.43443 * 0) + (0.08607 * 0.43779) + (0.20086 * 0) \\ +(0.27117 * 0) + (0.12713 * 0.43779) + (0.27117 * 0) + (0.15438 * 0) + \\ (0.27117 * 0) + (0.22804 * 0.78529) + (0.27117 * 0) + (0.20086 * 0) + \\ (0.15438 * 0) + (0.43443 * 0) \end{pmatrix}$$

$$= 0.272417918$$

Berikut merupakan hasil dari perhitungan *cosine similarity* yang ditunjukkan pada Tabel 4.12.

Tabel 4.12 Hasil *Cosine Similarity*

No	Term	*Cosine Similarity*				
		Dok 1	Dok 2	Dok 3	Dok 4	Dok 5
1	administrasi	0	0	0	0	0
2	agresif	0	0	0	0	0
3	android	0	0	0	0	0
4	aspi	0	0	0	0	0
5	atm	0	0	0	0	0
6	bank	0.04834	0.03768	0	0.08944	0
7	bobol	0	0	0	0	0
8	bri	0	0	0	0	0
9	bumn	0	0	0	0	0
10	chip	0	0	0	0	0
11	dana	0.05488	0.05566	0.10684	0	0
12	data	0	0	0	0	0
13	debit	0	0	0	0	0
14	ekonomi	0	0	0	0	0
15	hilang	0	0	0	0	0
16	jamin	0	0	0	0	0
17	kartu	0	0	0	0	0
18	komputer	0	0	0	0	0
19	korban	0	0	0	0	0
20	lapor	0	0	0	0	0
21	misteri	0	0	0	0	0
22	nasabah	0.19152	0.17908	0	0	0
23	nasional	0	0	0	0	0
24	negara	0	0	0	0	0
25	operasi	0	0	0	0	0
26	pendidikan	0	0	0	0	0
27	praktik	0	0	0	0	0
28	presiden	0	0	0	0	0
29	pso	0	0	0	0	0
30	raib	0	0	0	0	0
31	risiko	0	0	0	0	0
32	sekolah	0	0	0	0	0
33	sistem	0	0	0	0	0
34	skimming	0	0	0	0	0

Tabel 4.12 Hasil *Cosine Similarity* (lanjutan)

No	Term	*Cosine Similarity*				
		Dok 1	Dok 2	Dok 3	Dok 4	Dok 5
35	sma	0	0	0	0	0
36	teknologi	0	0	0	0	0
37	transaksi	0	0	0	0	0
38	uji	0	0	0	0	0
39	ujian	0	0	0	0	0
40	usbn	0	0	0	0	0
Jumlah		**0.29474**	**0.27242**	**0.10684**	**0.08944**	**0**

Kemudian setelah mendapatkan tabel *cosine similarity*, dilakukan pengurutan bobot dokumen (data latih) terhadap data uji dari yang paling relevan hingga tidak relevan. Berikut Tabel 4.13 menunjukkan hasil pengurutan bobotnya.

Tabel 4.13 Urutan Bobot Dokumen

Rank	Dokumen	Bobot		
1	Dok 1	0.29474	Dok. Relevan	Nilai V
2	Dok 2	0.27242		
3	Dok 3	0.10684	Dok. Tidak Relevan	
4	Dok 4	0.08944		
5	Dok 5	0		

Dokumen 1 memiliki nilai *cosine similarity* tertinggi (V) sehingga digunakan sebagai pembanding dengan nilai σ yang akan dilakukan pada tahapan proses selanjutnya. Dokumen 1 dan 2 dianggap dokumen relevan, sedangkan dokumen 3 dan 4 dianggap tidak relevan

4.3.4 *Extended Rocchio Relevance Feedback*

Pada proses perhitungan *Extended Rocchio* ini dilakukan dengan beberapa tahapan, yaitu *Query Modification, Relevance Feedback,* dan *Profile Modification*. 3 pendekatan tersebut nantinya digunakan untuk menghasilkan *query* tambahan, agar pencarian lebih spesifik.

4.3.4.1 *Query Modification*

Pada proses ini nilai σ diset dengan nilai yang tidak jauh dari penelitian Jordan Chris & Watters Carolyn (2004), serta nilai ini dapat diset dengan nilai random dan nantinya akan mempengaruhi hasil, karena nilai ini digunakan sebagai *threshold* untuk menentukan rumus modifikasi *query* atau tanpa modifikasi *query.*

$$\sigma = 0.25$$

Nilai tersebut digunakan untuk menentukan apakah *query* nantinya akan di modifikasi atau tidak. Nilai σ akan dibandingkan dengan nilai tertinggi dari bobot *similarity* pada dokumen yang relevan, dimana pada perhitungan ini dokumen

yang memiliki nilai tertinggi terletak pada dokumen 1. Berikut merupakan dokumen relevan yang ditentukan pada Tabel 4.14.

Tabel 4.14 Dokumen Relevan

Rank	Dokumen	Bobot
1	Dok 1	0.29474
2	Dok 2	0.27242
3	Dok 3	0.10684
4	Dok 4	0.08944
5	Dok 5	0

Pada dokumen relevan di atas menunjukkan bahwa bobot dokumen tertinggi bernilai lebih besar dari nilai σ (σ<V), maka sesuai dengan kondisi aturan pada *query modification*, *query* akan dilakukan modifikasi (Q_{mod}). Sedangkan pada *profile modification*, nantinya akan menggunakan rumus pada persamaan (2.9).

Selanjutnya dilakukan perhitungan dengan menghitung bobot rata-rata (*average weight)*, dimana perhitungan dilakukan pada term di tiap dokumen yang memiliki nilai selain nilai nol pada tabel hasil hitung normalisasi sebelumnya pada Tabel (4.11), yang dibagi menjadi 3 *Term Vector*, yaitu

1. *Term Vector P* (pada dokumen relevan, yaitu dokumen 1 dan 2),
2. *Term vector N* (pada dokumen tidak relevan, yaitu dokumen 3 dan 4), dan
3. *Term Vector F* (pada dokumen dengan *cosine similarity* bernilai 0, yaitu dokumen 5).

Berikut merupakan contoh perhitungan *average weight* pada *term* atm :

$$Avg\ Weight\ P = \frac{4.34432669}{1} = 4.34432669$$

Pada perhitungan di atas, yaitu pada term atm, dibagi dengan nilai 1, karena pada term atm hanya terdapat 1 nilai yang muncul atau selain nilai 0, yaitu pada dokumen 2, jika muncul kedua nilai selain nilai 0 maka akan dibagi dengan 2. Berikut merupakan hasil perhitungannya yang ditunjukkan pada Tabel 4.15.

Tabel 4.15 *Average Weight Term Vector* P

No	Wtd	Dokumen		Average Weight
		Dok 1	Dok 2	Term Vector P
1	administrasi	0	0	0
2	agresif	0	0	0
3	android	0	0	0
4	aspi	0	0.35280	0.35280
5	atm	0	0.43443	0.43443
6	bank	0.11042	0.08607	0.09824
7	bobol	0.26739	0	0.26739
8	bri	0.19806	0.20086	0.19946

Tabel 4.15 *Average Weight Term Vector* P (lanjutan)

No	Wtd	Dokumen		Average Weight
		Dok 1	Dok 2	Term Vector P
9	bumn	0	0	0
10	chip	0	0.27117	0.27117
11	dana	0.12536	0.12713	0.12625
12	data	0.34788	0	0.34788
13	debit	0	0.27117	0.27117
14	ekonomi	0	0	0
15	hilang	0.15223	0.15438	0.15331
16	jamin	0.34788	0	0.34788
17	kartu	0	0.27117	0.27117
18	komputer	0	0	0
19	korban	0.26739	0	0.26739
20	lapor	0.26739	0	0.26739
21	misteri	0.26739	0	0.26739
22	nasabah	0.24389	0.22804	0.23596
23	nasional	0	0	0
24	negara	0.26739	0	0.26739
25	operasi	0	0	0
26	pendidikan	0	0	0
27	praktik	0	0.27117	0.27117
28	presiden	0	0	0
29	pso	0	0	0
30	raib	0.15223	0.20086	0.17655
31	risiko	0	0	0
32	sekolah	0	0	0
33	sistem	0	0.15438	0.15438
34	skimming	0	0.43443	0.43443
35	sma	0	0	0
36	teknologi	0.26739	0	0.26739
37	transaksi	0.39497	0	0.39497
38	uji	0	0	0
39	ujian	0	0	0
40	usbn	0	0	0

Berikut merupakan contoh perhitungan *average weight* pada *term* atm :

$$Avg\ Weight\ N = 0$$

Pada term atm bernilai 0 untuk *Average Weight Term Vector* N, karena pada term tersebut memiliki nilai 0 pada dokumen 3 dan 4, sehingga tidak dilakukan

perhitungan rata-rata. Berikut merupakan hasil perhitungannya yang ditunjukkan pada Tabel 4.16.

Tabel 4.16 *Average Weight Term Vector* N

No	Wtd	Dokumen		Average Weight
		Dok 3	Dok 4	Term Vector N
1	administrasi	0.59098	0	0.59098
2	agresif	0	0.34884	0.34884
3	android	0	0	0
4	aspi	0	0	0
5	atm	0	0	0
6	bank	0	0.20429	0.20429
7	bobol	0	0	0
8	bri	0	0	0
9	bumn	0	0.34884	0.34884
10	chip	0	0	0
11	dana	0.24404	0	0.24404
12	data	0	0	0
13	debit	0	0	0
14	ekonomi	0	0.34884	0.34884
15	hilang	0	0	0
16	jamin	0	0	0
17	kartu	0	0	0
18	komputer	0	0	0
19	korban	0	0	0
20	lapor	0	0	0
21	misteri	0	0	0
22	nasabah	0	0	0
23	nasional	0	0.34884	0.34884
24	negara	0	0	0
25	operasi	0	0	0
26	pendidikan	0	0	0
27	praktik	0	0	0
28	presiden	0	0.45385	0.45385
29	pso	0.76889	0	0.76889
30	raib	0	0	0
31	risiko	0	0.51528	0.51528
32	sekolah	0	0	0
33	sistem	0	0	0
34	skimming	0	0	0
35	sma	0	0	0
36	teknologi	0	0	0

Tabel 4.16 *Average Weight Term Vector N* (lanjutan)

No	Wtd	Dokumen		Average Weight
		Dok 3	Dok 4	Term Vector N
37	transaksi	0	0	0
38	uji	0	0	0
39	ujian	0	0	0
40	usbn	0	0	0

Berikut merupakan contoh perhitungan *average weight* pada *term* atm :

$$Avg\ Weight\ F = 0$$

Pada term atm bernilai 0 untuk *Average Weight Term Vector* F, karena pada term tersebut memiliki nilai 0 pada dokumen 5, sehingga tidak dilakukan perhitungan rata-rata. Berikut merupakan hasil perhitungannya yang ditunjukkan pada Tabel 4.17.

Tabel 4.17 *Average Weight Term Vector* F

No	Wtd	Dokumen	Average Weight
		Dok 5	Term Vector F
1	administrasi	0	0
2	agresif	0	0
3	android	0.39812	0.39812
4	aspi	0	0
5	atm	0	0
6	bank	0	0
7	bobol	0	0
8	bri	0	0
9	bumn	0	0
10	chip	0	0
11	dana	0	0
12	data	0	0
13	debit	0	0
14	ekonomi	0	0
15	hilang	0	0
16	jamin	0	0
17	kartu	0	0
18	komputer	0.26952	0.26952
19	korban	0	0
20	lapor	0	0
21	misteri	0	0
22	nasabah	0	0
23	nasional	0	0

Tabel 4.17 *Average Weight Term Vector F* (lanjutan)

No	Wtd	Dokumen	Average Weight
		Dok 5	Term Vector F
24	negara	0	0
25	operasi	0.26952	0.26952
26	pendidikan	0.26952	0.26952
27	praktik	0	0
28	presiden	0	0
29	pso	0	0
30	raib	0	0
31	risiko	0	0
32	sekolah	0.39812	0.39812
33	sistem	0.15345	0.15345
34	skimming	0	0
35	sma	0.35066	0.35066
36	teknologi	0	0
37	transaksi	0	0
38	uji	0.35066	0.35066
39	ujian	0.26952	0.26952
40	usbn	0.35066	0.35066

4.3.4.2 *Relevance Feedback*

Pada proses ini dilakukan untuk menentukan nilai pada parameter *term vector* P, N, dan F, berdasarkan dari hasil perhitungan *Average Weight* pada *Term vector* P, N, dan F sebelumnya pada Tabel (4.15), (4.16), dan (4.17).

1. *Term Vector P*

Term Vector ini mengambil nilai dari perhitungan *Average Weight* pada *term vector P* yang sebelumnya telah dihitung pada Tabel 4.15, namun pada bagian ini akan dipilah kembali, karena mengambil nilai selain dari *query*, Q asli.
Berikut merupakan contoh perhitungan *Term Vector P* pada *term* atm :

$$Term\ Vector\ P = 0.352801978,$$

Bernilai tetap, karena pada term atm tidak termasuk pada *query* sehingga bernilai tetap.

Berikut merupakan hasil perhitungannya yang ditunjukkan pada Tabel 4.18.

Tabel 4.18 *Term Vector* P

No	Term	*Avg Weight Term Vector* P	Query	*Term Vector P*
1	administrasi	0	0	0
2	agresif	0	0	0
3	android	0	0	0

Tabel 4.18 *Term Vector* P (lanjutan)

No	Term	*Avg Weight Term Vector* P	Query	*Term Vector P*
4	aspi	0.35280	0	0.35280
5	atm	0.43443	0	0.43443
6	bank	0.09824	0.43779	0
7	bobol	0.26739	0	0.26739
8	bri	0.19946	0	0.19946
9	bumn	0	0	0
10	chip	0.27117	0	0.27117
11	dana	0.12627	0.43779	0
12	data	0.34788	0	0.34788
13	debit	0.27117	0	0.27117
14	ekonomi	0	0	0
15	hilang	0.15331	0	0.15331
16	jamin	0.34788	0	0.34788
17	kartu	0.27117	0	0.27117
18	komputer	0	0	0
19	korban	0.26739	0	0.26739
20	lapor	0.26739	0	0.26739
21	misteri	0.26739	0	0.26739
22	nasabah	0.23596	0.78529	0
23	nasional	0	0	0
24	negara	0.26739	0	0.26739
25	operasi	0	0	0
26	pendidikan	0	0	0
27	praktik	0.27117	0	0.27117
28	presiden	0	0	0
29	pso	0	0	0
30	raib	0.17655	0	0.17655
31	risiko	0	0	0
32	sekolah	0	0	0
33	sistem	0.15438	0	0.15438
34	skimming	0.43443	0	0.43443
35	sma	0	0	0
36	teknologi	0.26739	0	0.26739
37	transaksi	0.39497	0	0.39497
38	uji	0	0	0
39	ujian	0	0	0
40	usbn	0	0	0

2. *Term Vector* N

Term Vector ini mengambil nilai dari perhitungan *Average Weight* pada *term vector N* yang sebelumnya telah dihitung pada Tabel 4.16, namun pada bagian ini akan dipilah kembali, karena mengambil nilai selain dari *query*, Q asli.
Berikut merupakan contoh perhitungan *Term Vector N* pada *term* atm :

$$Term\,Vector\,N = 0,$$

Bernilai tetap, karena pada term atm tidak termasuk pada *query* sehingga bernilai tetap.

Berikut merupakan hasil perhitungannya yang ditunjukkan pada Tabel 4.19.

Tabel 4.19 *Term Vector* N

No	Term	*Avg Weight Term Vector* N	Query	*Term Vector N*
1	administrasi	0.59098	0	0.59098
2	agresif	0.34884	0	0.34884
3	android	0	0	0
4	aspi	0	0	0
5	atm	0	0	0
6	bank	0.20429	0.43779	0
7	bobol	0	0	0
8	bri	0	0	0
9	bumn	0.34884	0	0.34884
10	chip	0	0	0
11	dana	0.24403	0.43779	0
12	data	0	0	0
13	debit	0	0	0
14	ekonomi	0.34884	0	0.34884
15	hilang	0	0	0
16	jamin	0	0	0
17	kartu	0	0	0
18	komputer	0	0	0
19	korban	0	0	0
20	lapor	0	0	0
21	misteri	0	0	0
22	nasabah	0	0.78529	0
23	nasional	0.34884	0	0.34884
24	negara	0	0	0
25	operasi	0	0	0
26	pendidikan	0	0	0
27	praktik	0	0	0
28	presiden	0.45385	0	0.45385

Tabel 4.19 *Term Vector* N (lanjutan)

No	Term	*Avg Weight Term Vector* N	Query	*Term Vector N*
29	pso	0.76889	0	0.76889
30	raib	0	0	0
31	risiko	0.51528	0	0.51528
32	sekolah	0	0	0
33	sistem	0	0	0
34	skimming	0	0	0
35	sma	0	0	0
36	teknologi	0	0	0
37	transaksi	0	0	0
38	uji	0	0	0
39	ujian	0	0	0
40	usbn	0	0	0

3. *Term Vector* F

Term Vector ini mengambil nilai dari perhitungan *Average Weight* pada *term vector N* yang sebelumnya telah dihitung pada Tabel 4.17, namun pada bagian ini akan dipilah kembali, karena mengambil nilai selain dari *query*, Q asli, serta selain nilai pada P dan N.
Berikut merupakan contoh perhitungan *Term Vector F* pada *term* atm :

$$Term\ Vector\ F = 0,$$

Bernilai tetap, karena pada term atm tidak termasuk pada *query(Q)*, P, dan N sehingga bernilai tetap. Sedangkan pada *term* 'sistem' memiliki nilai:

$$Term\ Vector\ F = 0,$$

Diubah menjadi nilai 0, karena walaupun *term* tersebut tidak ada pada *query*, Q asli, namun *term* tersebut ada pada nilai *term vector P* sehingga nilai *term vector F* harus diubah ke nilai 0, karena *term vector F* berisi nilai selain dari P, N, dan Q.

Berikut merupakan hasil perhitungannya yang ditunjukkan pada Tabel 4.20.

Tabel 4.20 *Term Vector* F

No	Term	Query	*Avg Weight Term Vector* F	Term Vector		
				Term Vector P	Term Vector N	Term Vector F
1	administrasi	0	0	0	0.59098	0
2	agresif	0	0	0	0.34884	0
3	android	0	0.39812	0	0	0.39812
4	aspi	0	0	0.35280	0	0
5	atm	0	0	0.43443	0	0

Tabel 4.20 *Term Vector* F (lanjutan)

No	Term	Query	*Avg Weight Term Vector* F	Term Vector		
				Term Vector P	Term Vector N	Term Vector F
6	bank	0.43779	0	0	0	0
7	bobol	0	0	0.26739	0	0
8	bri	0	0	0.19946	0	0
9	bumn	0	0	0	0.34884	0
10	chip	0	0	0.27117	0	0
11	dana	0.43779	0	0	0	0
12	data	0	0	0.34788	0	0
13	debit	0	0	0.27117	0	0
14	ekonomi	0	0	0	0.34884	0
15	hilang	0	0	0.15331	0	0
16	jamin	0	0	0.34788	0	0
17	kartu	0	0	0.27117	0	0
18	komputer	0	0.26952	0	0	0.26952
19	korban	0	0	0.26739	0	0
20	lapor	0	0	0.26739	0	0
21	misteri	0	0	0.26739	0	0
22	nasabah	0.78529	0	0	0	0
23	nasional	0	0	0	0.34884	0
24	negara	0	0	0.26739	0	0
25	operasi	0	0.26952	0	0	0.26952
26	pendidikan	0	0.26952	0	0	0.26952
27	praktik	0	0	0.27117	0	0
28	presiden	0	0	0	0.45385	0
29	pso	0	0	0	0.76889	0
30	raib	0	0	0.17655	0	0
31	risiko	0	0	0	0.51528	0
32	sekolah	0	0.39812	0	0	0.39812
33	sistem	0	0.15345	0.15438	0	0
34	skimming	0	0	0.43443	0	0
35	sma	0	0.35066	0	0	0.35066
36	teknologi	0	0	0.26739	0	0
37	transaksi	0	0	0.39497	0	0
38	uji	0	0.35066	0	0	0.35066
39	ujian	0	0.26952	0	0	0.26952
40	usbn	0	0.35066	0	0	0.35066

4.3.4.3 *Profile Modification*

Pada proses ini dilakukan dengan acuan yang berasal dari *query modification* pada tahap sebelumnya. Seperti yang diketahui nilai *term vector* lebih besar dari nilai yang ditetapkan pada nilai σ.

$$\sigma = 0.25\,, \qquad \sigma < V$$

Sehingga modifikasi *query* dilakukan dengan rumus persamaan (2.9). Sedangkan jika nilai $\sigma > V$, maka akan menggunakan persamaan rumus (2.8). Sebelumnya ditentukan terlebih dahulu nilai α, β, γ, dan Δ secara random.

$$\alpha = 1.25, \beta = 0.79, \gamma = 0.28, dan\ \Delta = 0.54$$

Nilai-nilai parameter di atas ditentukan dengan nilai random dengan nilai yang diset tidak jauh dari nilai pada penelitian Jordan Chris & Watters Carolyn (2004) dan berikut merupakan contoh perhitungan pada *term* atm:

$$V = ((1.25 * 0) + (0.79 * 0.4344327)) - ((0.28 * 0) + (0.54 * 0)) = 0.34320$$

Term	Query	Term Vector P	Term Vector N	Term Vector F
atm	0	0.43443	0	0

Berikut hasil perhitungannya ditunjukkan pada Tabel 4.21.

Tabel 4.21 Hasil Perhitungan V

No	Term	V	
1	administrasi	-0.16547	0
2	agresif	-0.09768	0
3	android	-0.21498	0
4	aspi	0.27871	0.27871
5	atm	0.34320	0.34320
6	bank	0.54724	0.54724
7	bobol	0.21124	0.21124
8	bri	0.15757	0.15757
9	bumn	-0.09768	0
10	chip	0.21423	0.21423
11	dana	0.54724	0.54724
12	data	0.27483	0.27483
13	debit	0.21423	0.21423
14	ekonomi	-0.09768	0
15	hilang	0.12111	0.12111
16	jamin	0.27483	0.27483
17	kartu	0.21423	0.21423
18	komputer	-0.14554	0

Tabel 4.21 Hasil Perhitungan V (lanjutan)

No	Term	V	
19	korban	0.21124	0.21124
20	lapor	0.21124	0.21124
21	misteri	0.21124	0.21124
22	nasabah	0.98161	0.98161
23	nasional	-0.09768	0
24	negara	0.21124	0.21124
25	operasi	-0.14554	0
26	pendidikan	-0.14554	0
27	praktik	0.21423	0.21423
28	presiden	-0.12708	0
29	pso	-0.21529	0
30	raib	0.13947	0.13947
31	risiko	-0.14428	0
32	sekolah	-0.21498	0
33	sistem	0.12196	0.12196
34	skimming	0.34320	0.34320
35	sma	-0.18936	0
36	teknologi	0.21124	0.21124
37	transaksi	0.31203	0.31203
38	uji	-0.18936	0
39	ujian	-0.14554	0
40	usbn	-0.18936	0

Setelah nilai V dihasilkan, lalu dilakukan perurutan untuk nilai bobot terbesar hingga terkecil, dimana nilai bobot terbesar akan dijadikan *query* tambahan dari *query* aslinya. Berikut pada Tabel 2.22 menunjukkan hasil perurutannya.

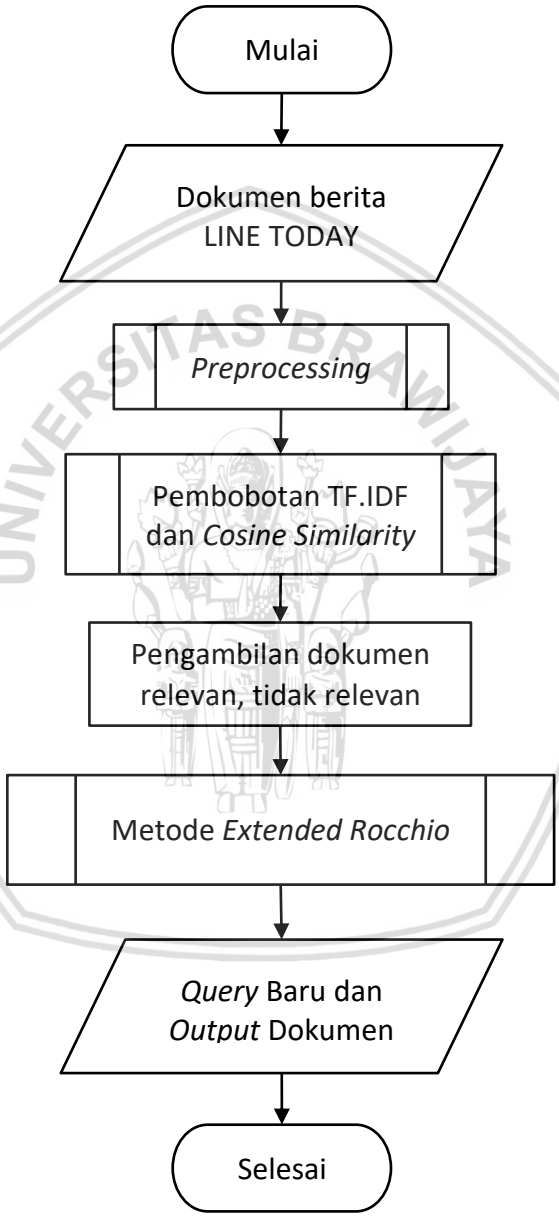
Tabel 4.22 Hasil Rank V

RANK		
nasabah	0.98160910	1
bank	0.54724015	2
dana	0.54724015	3
atm	0.34320181	4
skimming	0.34320181	5
transaksi	0.31202587	6
aspi	0.27871356	7
data	0.27482850	8
jamin	0.27482850	9
debit	0.21422532	10

Pada *term* dana, nasabah, dan bank, merupakan *query* asli sehingga tambahan *query* akan diambil dari rank nomor 4-10.

4.3.5 Diagram Alir Sistem

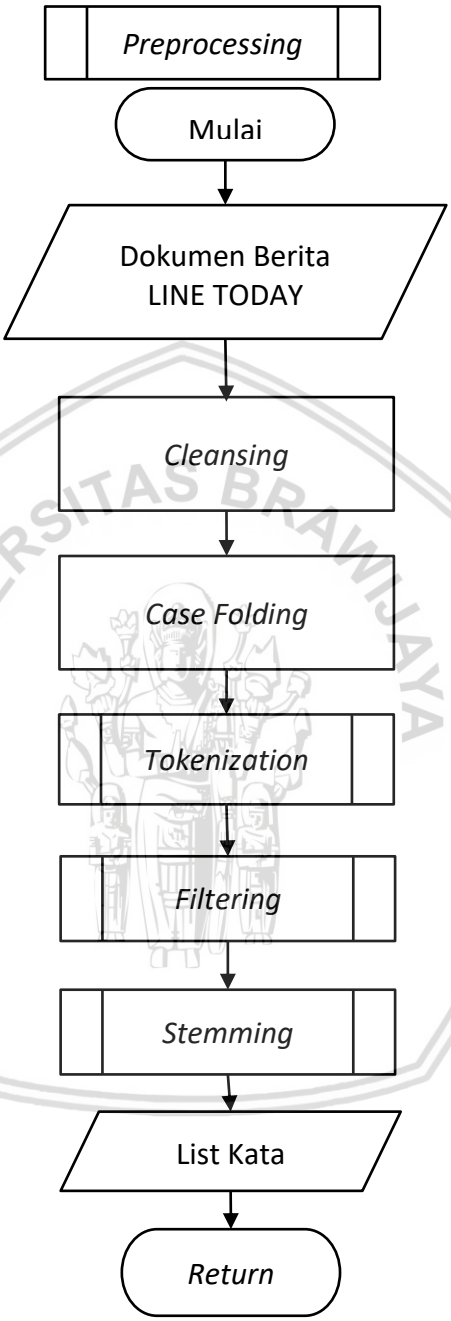
Berikut ini merupakan Gambar 4.1, menjelaskan diagram alir sistem proses yang dilakukan, diantaranya *preprocessing, term wighting, normalisasi* dan *cosine similarity,* dan metode *extended rocchio relevance feedback.*



Gambar 4.1 Diagram Alir Sistem

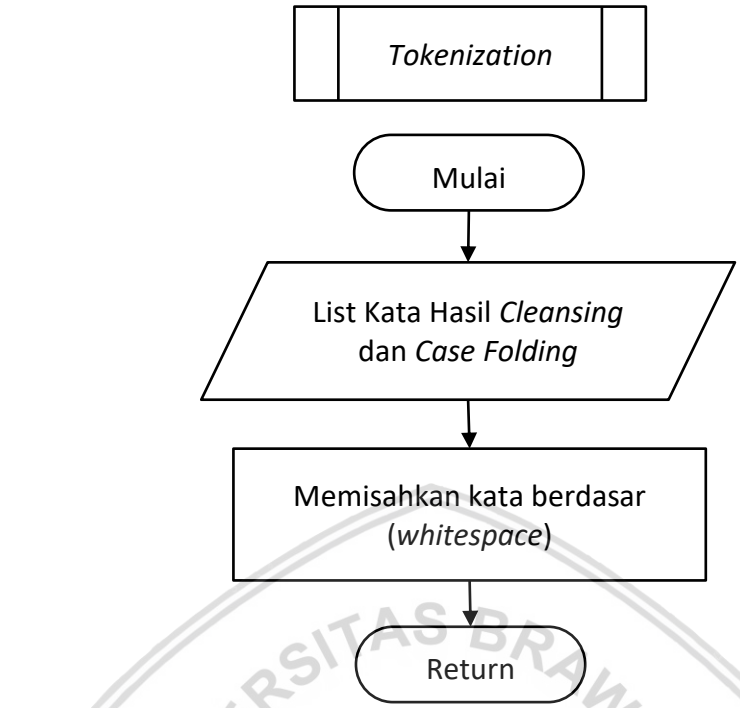
Gambar 4.2 di bawah ini menunjukkan gambar diagram alir dari proses *preprocessing*, dimana proses tersebut digunakan dengan melalui beberapa tahapan, yaitu dengan dilakukan *case folding* untuk mengubah huruf kapital menjadi huruf kecil *cleansing,* untuk membuang keseluruhan karakter selain huruf,

tokenization untuk memisahkan tiap kata yang dipisah dari *whitespace*, *filtering* untuk menyaring kata-kata yang tidak penting atau menghapus kata-kata yang berada dalam list *stopword*, serta *stemming* untuk merubah seluruh kata menjadi kata dasar.



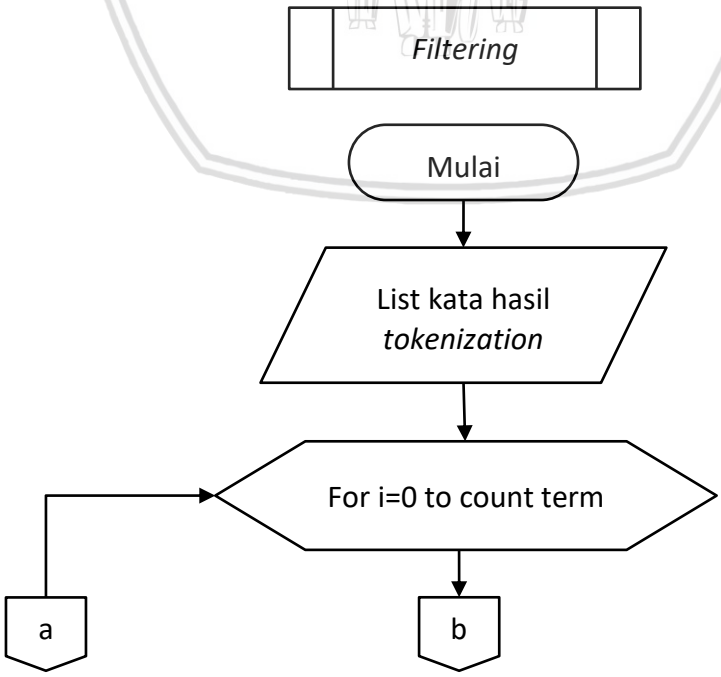
Gambar 4.2 Diagram Alir *Preprocessing*

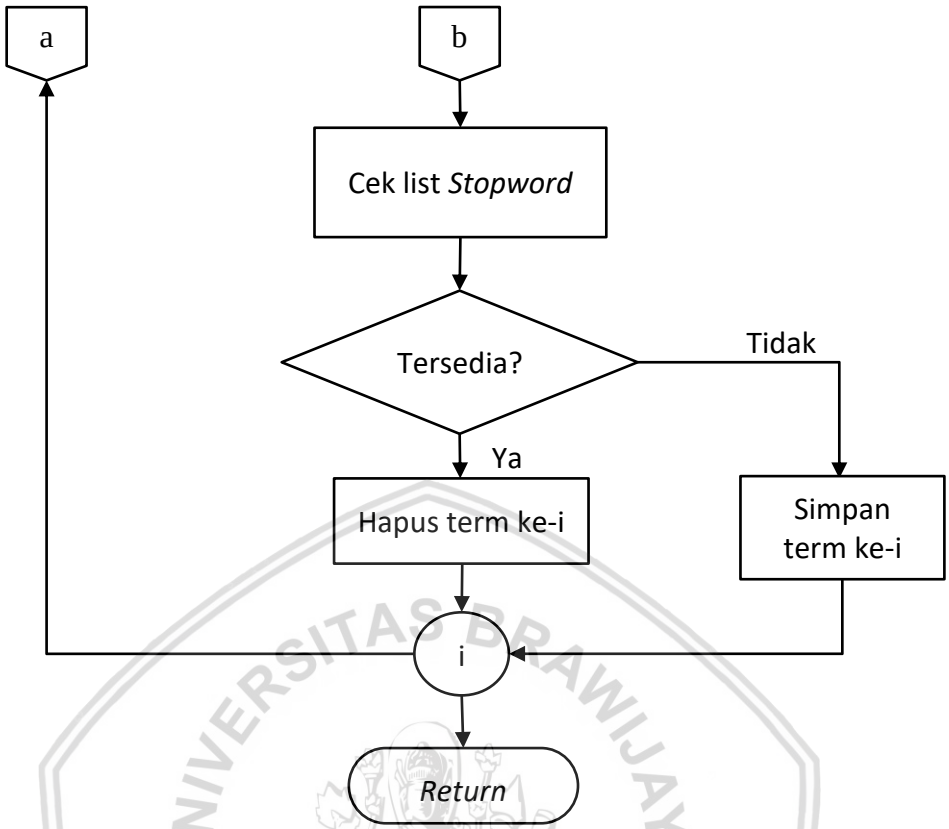
Pada Gambar 4.3 menggambarkan diagram alir pada proses *tokenization*, yaitu proses memisahkan tiap kata yang dipisahkan oleh *whitespace.* Selain itu, karakter-karakter selain huruf juga akan dihapuskan.



Gambar 4.3 Diagram Alir *Tokenization*

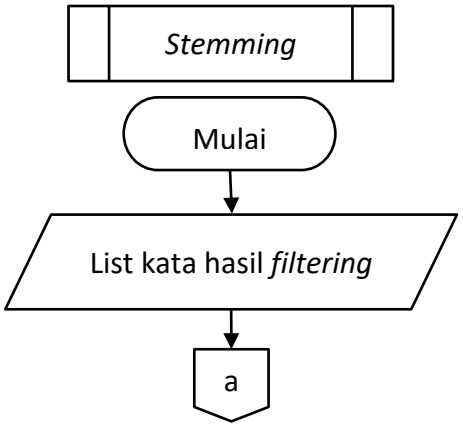
Selanjutnya pada Gambar 4.4 akan menjelaskan diagram alir dari proses *filtering*, dimana pada tahapan ini digunakan untuk mendapatkan dokumen dengan kata yang telah di *filtering* atau disaring dengan menghapus kata dari list *stopword.* Kata pada dokumen yang ada pada list *stopword* akan dihilangkan, karena merupakan kata yang tidak penting. List dari *stopword* dapat di lihat pada Lampiran A.1.

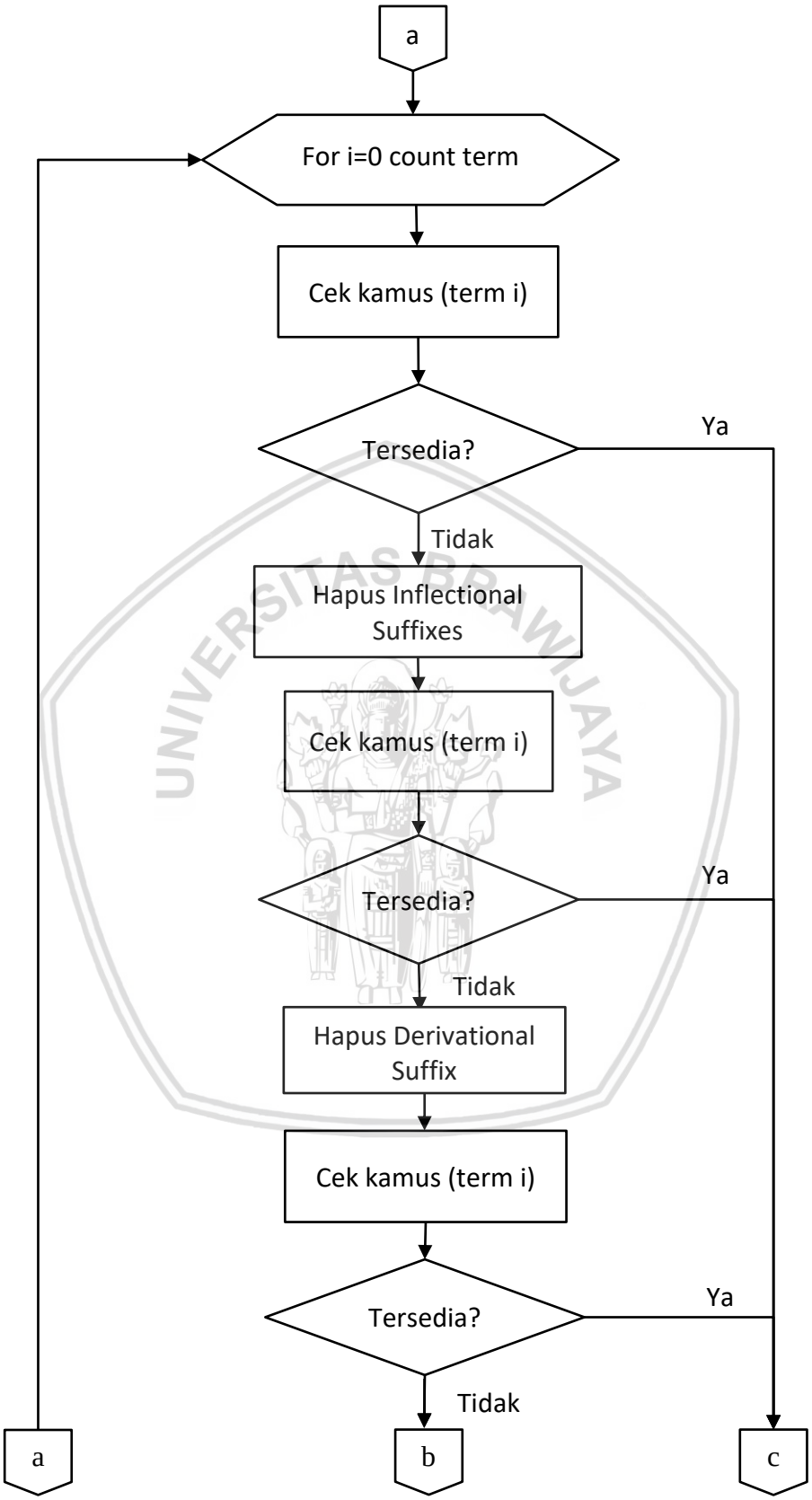


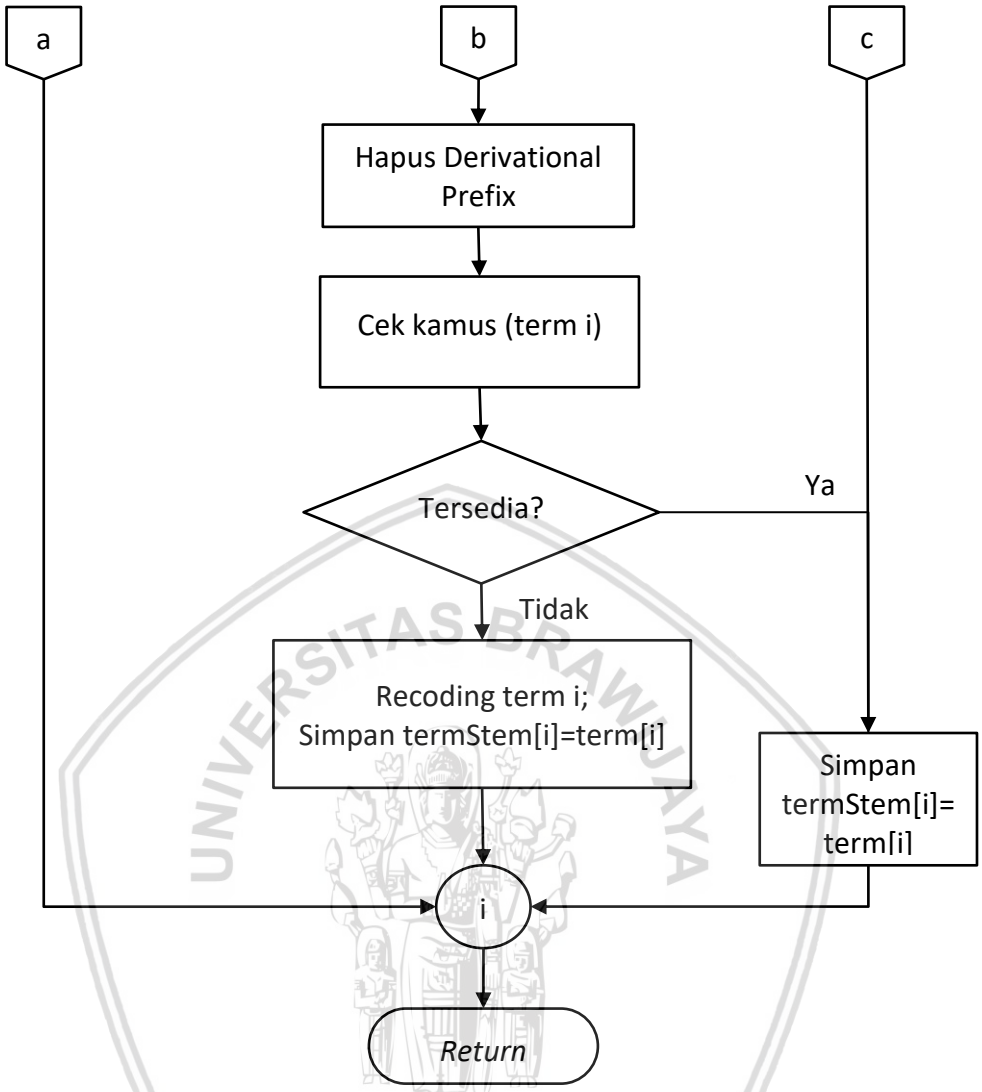


Gambar 4.4 Diagram Alir *Filtering*

Gambar 4.5 menggambarkan diagram alir *stemming*, dimana *stemming* merupakan tahapan terakhir dari proses *preprocessing*. Proses ini mengubah tiap term atau kata menjadi kata dasar. Proses *stemming* yang dimanfaatkan pada skripsi ini memanfaatkan algortima dari Nazief-Adriani. Proses ini diawali dengan menghapus *Infectional Suffixes*, *Dericational Suffix, Derivational Prefix,* dan di tiap tahapan akan dilakukan pengecekan tiap kata pada kata dasar yang digunakan. Jika tahapan tersebut telah dilakukan dan belum ditemukan kata dasar di dalam kamus, maka akan dikembalikan ke kata aslinya, yaitu sebelum dilakukan proses *stemming.*



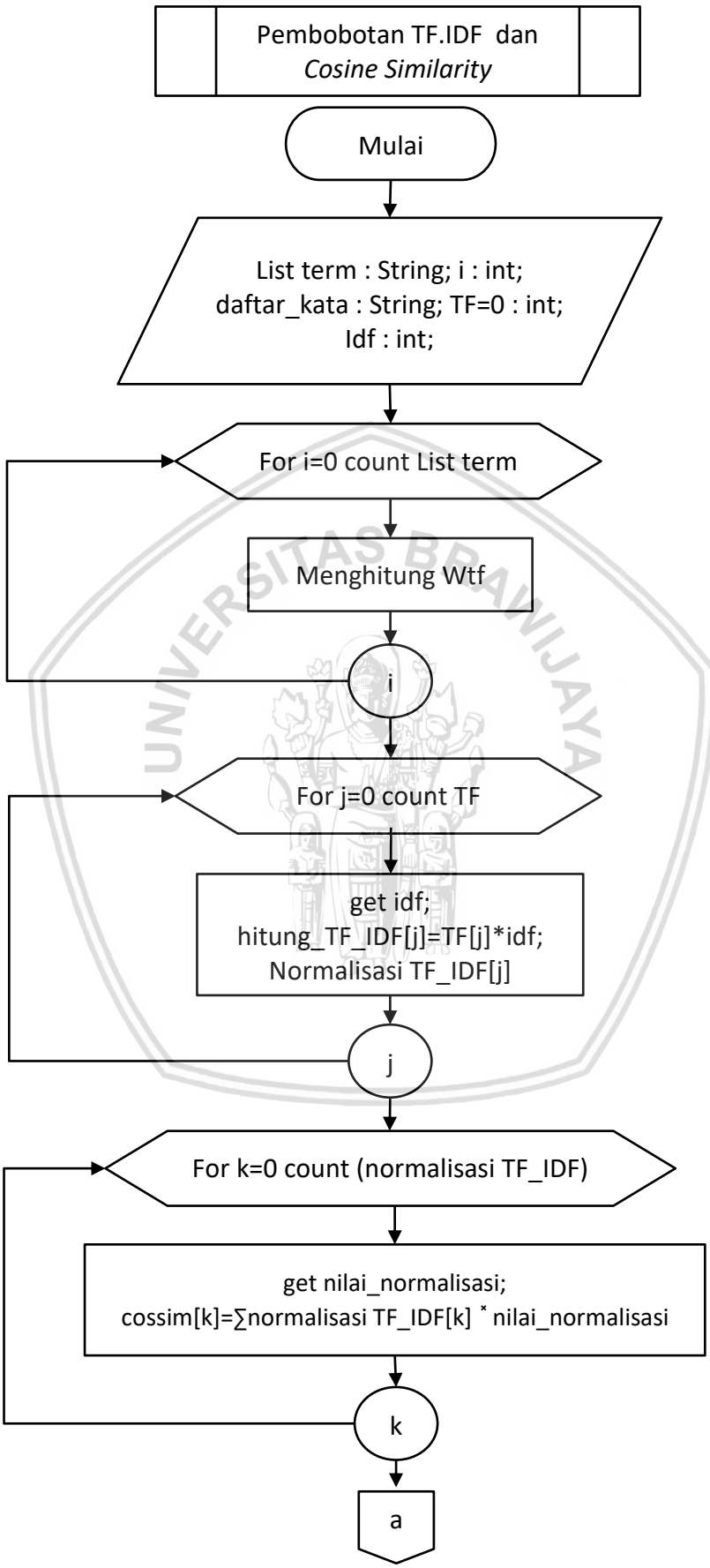

a
For i=0 count term
Cek kamus (term i)
Tersedia?
Ya
Tidak
Hapus Inflectional Suffixes
Cek kamus (term i)
Tersedia?
Ya
Tidak
Hapus Derivational Suffix
Cek kamus (term i)
Tersedia?
Ya
Tidak
a
b
c

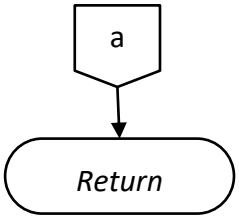


Gambar 4.5 Diagram Alir *Stemming*

Gambar 4.6 akan menjelaskan tentang gambaran dari diagram alir pembobotan TF.IDF dan *Cosine Similarity*. Proses ini dilakukan setelah proses *Stemming* telah selesai dilakukan, dimana tiap *term* dalam *stemming* akan dilakukan pembobotan term atau *term weighting*. Proses *term weighting* diproses dengan pembobotan Wtf dari hasil TF (*term frequency).* Proses IDF didapat dari data latih yang dihasilkan dari perkalian dengan hasil dari pembobotan TF. Setelah itu, dilakukan proses TF.IDF yang kemudian dinormalisasikan sebelum dilakukan proses *cosine similarity*.

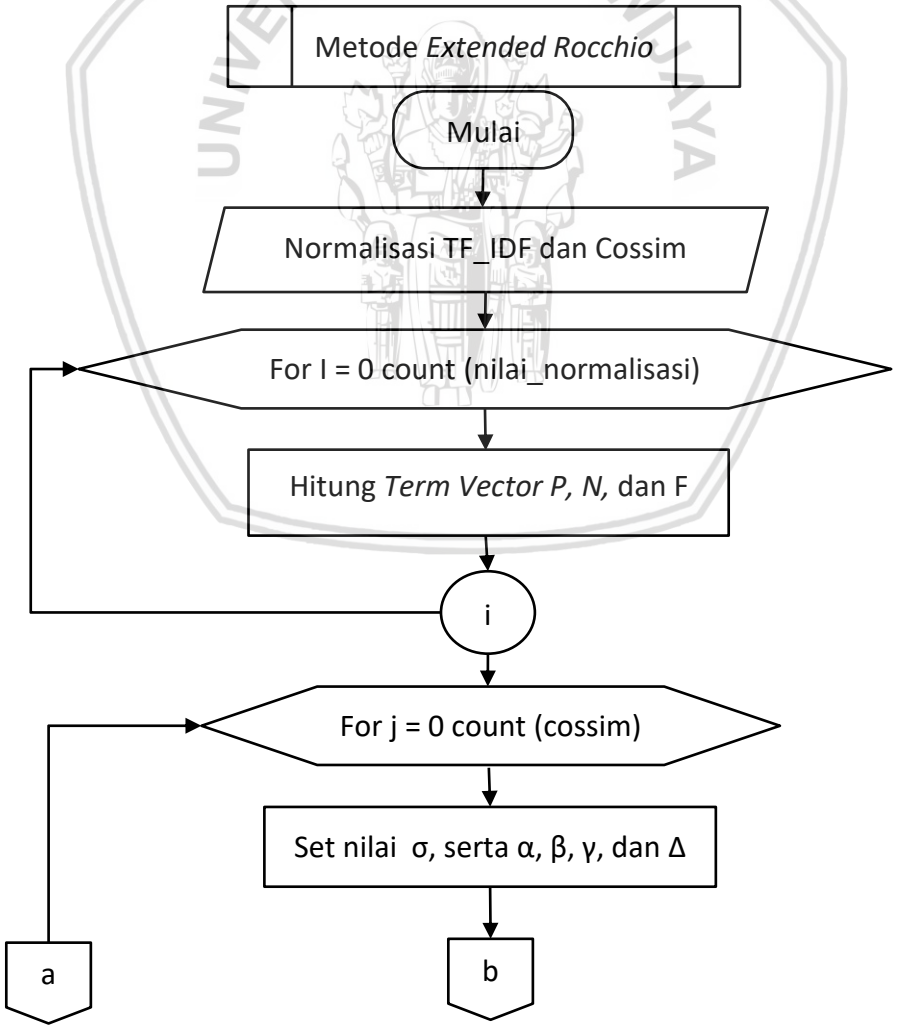
Perhitungan *cosine similarity* dilakukan pada data uji terhadap data latih. Sehingga akan ditemukan derajat kemiripan antara kedua data tersebut. Berikut merupakan diagram alir pembobotannya ditunjukkan pada Gambar 4.6.

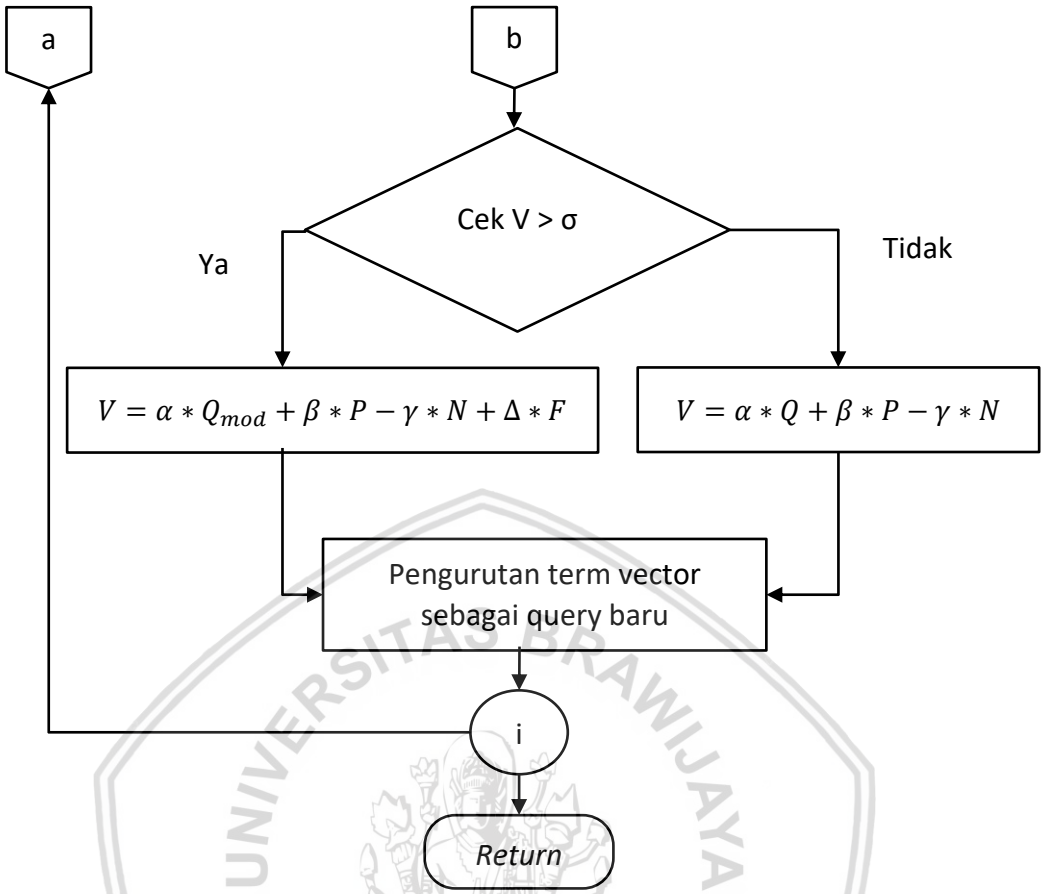

Pembobotan TF.IDF dan
Cosine Similarity
Mulai
List term : String; i : int;
daftar_kata : String; TF=0 : int;
Idf : int;
For i=0 count List term
Menghitung Wtf
i
For j=0 count TF
get idf;
hitung_TF_IDF[j]=TF[j]*idf;
Normalisasi TF_IDF[j]
j
For k=0 count (normalisasi TF_IDF)
get nilai_normalisasi;
cossim[k]=∑normalisasi TF_IDF[k] * nilai_normalisasi
k
a



Gambar 4.6 Diagram Alir Pembobotan TF.IDF dan *Cosine Similarity*

Gambar 4.7 menunjukkan diagram alir dari perhitungan Metode *Extended Rocchio Relevance Feedback,* dimana dimulai dengan *query* modification, yaitu menentukan nilai tho sebagai *threshold* yang akan dibandingkan dengan nilai *term vector V,* kemudian menghitung *relevance feedback* dengan mencari rata-rata dari *term weight* untuk tiap *term vector P, N,* dan F, serta menghitung *profile modification,* yang dilakukan dengan menggunakan rumus tertentu, dimana acuan yang digunakan ialah ada pada proses *query modification.* Maksudnya, ialah jika nilai tho lebih besar maupun lebih ekcil dari V, maka akan menggunakan rumus tertentu.





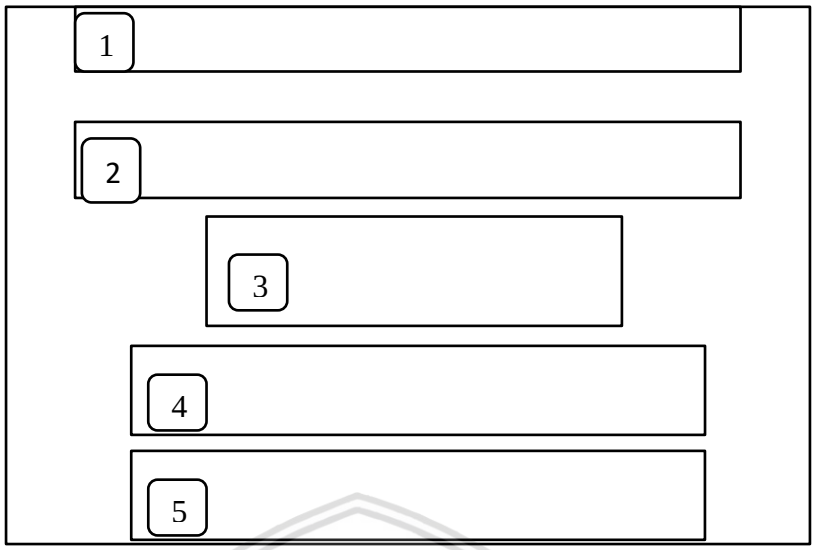
Gambar 4.7 Diagram Alir *Metode Extended Rocchio*

4.4 Perancangan Antarmuka

Pada perancangan ini digunakan untuk memahami hubungan atau interaksi antara pengguna dengan sistem. Berikut merupakan rancangan antarmuka yang telah dibangun, diantaranya:

4.4.1 Perancangan Antarmuka Halaman Awal

Pada halaman awal ini terdapat beberapa pilihan menu *button* dalam mengakses pengujian *query,* pengujian nilai parameter, hingga hasil pencarian dengan *query* tambahan. Terdapat diantaranya 4 menu pilihan, yaitu pengujian (perhitungan *cosine similarity),* pengujian nilai, hasil *query* tambahan, dan hasil algoritme *rocchio relevance feedback.* Berikut merupakan rancangan antarmuka pada halaman awal yang ditunjukkan pada Gambar 4.8.



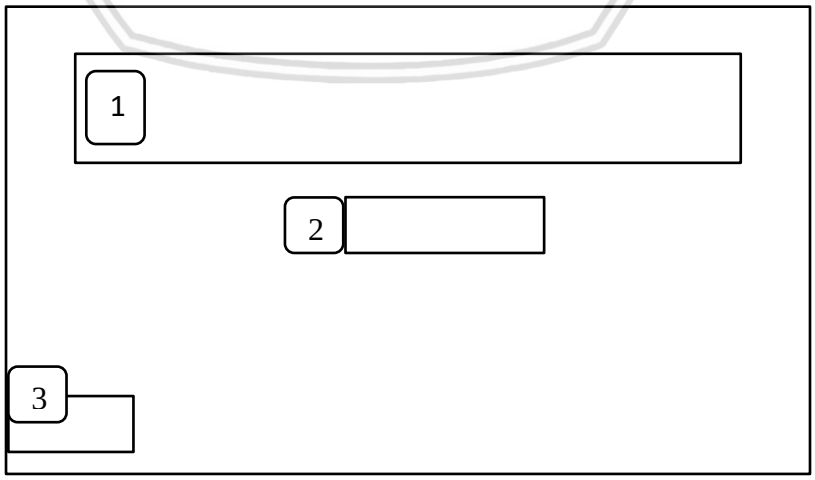
Gambar 4.8 Perancangan Antarmuka Halaman Awal

Keterangan:

1. Judul Sistem
2. Pengujian (Perhitungan *Cosine Similarity),* digunakan
3. Pengujian Nilai
4. Hasil *Query* Tambahan
5. Hasil Algoritme *Rocchio Relevance Feedback*

4.4.2 Perancangan Antarmuka Halaman Pengujian *Cosine Similarity*

Pada halaman ini menampilkan *box* untuk menginputkan *query* asli (25 data uji) tanpa ada tambahan *query*. Berikut merupakan perancangannya ditunjukkan pada Gambar 4.9.



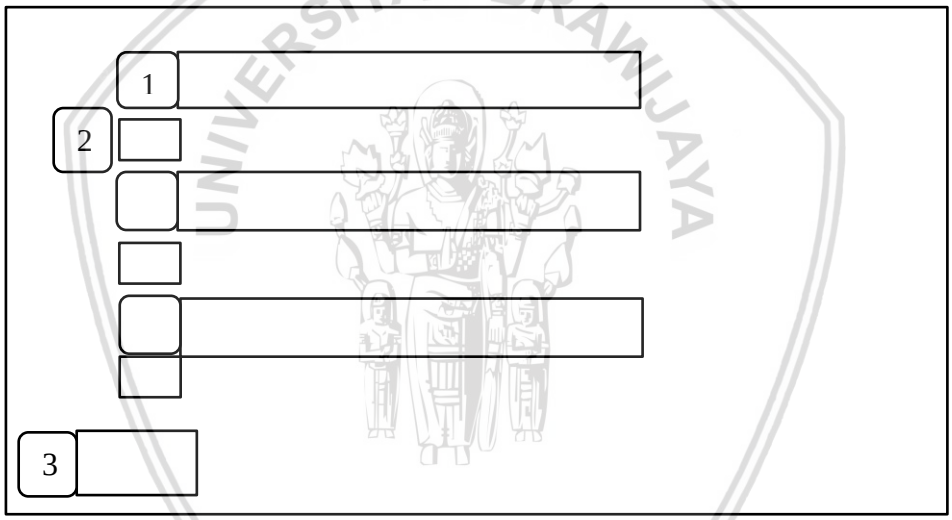
Gambar 4.9 Perancangan Antarmuka Pengujian *Cosine Similarity*

Keterangan:

1. Tempat Inputan *Query*
2. Tombol ‘Masukkan’, untuk memproses query hingga menampilkan hasil perangkingan dokumen
3. Tombol ‘Back’, untuk kembali ke halaman sebelumnya

4.4.3 Perancangan Antarmuka Halaman Hasil Pencarian Pengujian *Cosine Similarity*

Pada halaman ini merupakan hasil dari pencarian yang dilakukan di halaman Pengujian *Cosine Similarity*, dimana sebelumnya *query* (25 data uji) diinputkan di dalamnya sehingga menghasilkan hasil perangkingan dokumen yang sesuai dengan *query* yang telah terinputkan. Di tiap dokumen yang ditampilkan akan disediakan checkbox, guna untuk pengguna menilai tiap dokumen, apakah termasuk relevan atau tidak. Bila relevan, maka dokumen akan dicentang.Berikut ditunjukkan pada Gambar 4.10



Gambar 4.10 Perancangan Antarmuka Halaman Hasil Pencarian Pengujian *Cosine Similarity*

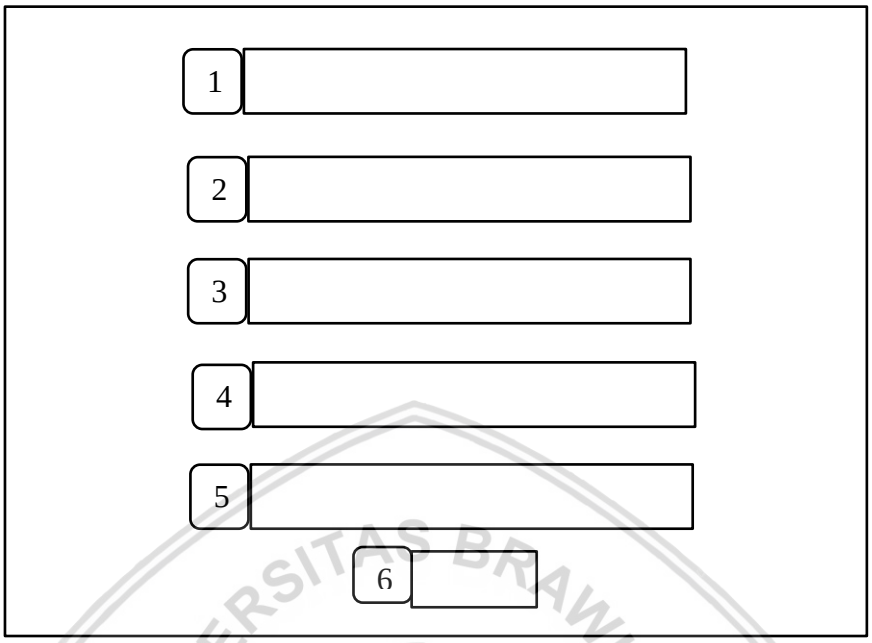
Keterangan:

1. Dokumen hasil dari pencarian *query*
2. *Checkbox* untuk menilai dokumen perangkingan hasil pencarian. Jika centang diberikan pada dokumen maka dokumen tersebut termasuk dokumen relevan dan muncul huruf ‘R’ disebelah *checkbox*
3. Tombol ‘Back’, untuk kembali ke halaman sebelumnya

4.4.4 Perancangan Antarmuka Halaman Pengujian Nilai

Pada halaman pengujian nilai, akan menampilkan hasil pengujian yang dilakukan pada beberapa parameter dengan nilai yang telah diset. Pengujian

terdiri dari parameter, *tho, alpha, beta, gama*, dan *delta* Berikut merupakan perancangannya ditunjukkan pada Gambar 4.11.



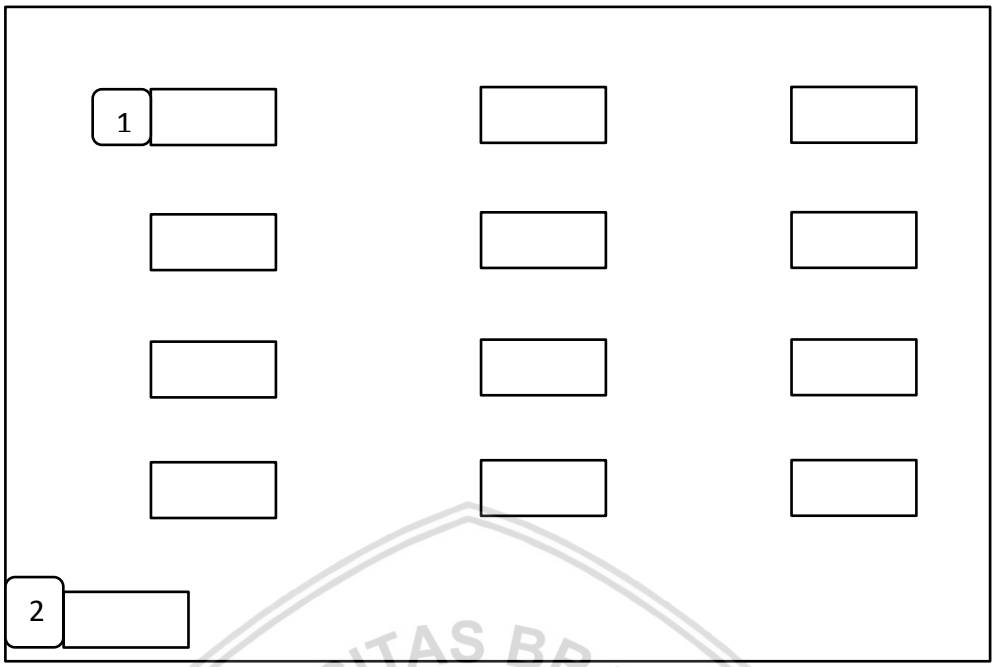
Gambar 4.11 Perancangan Antarmuka Halaman Pengujian Nilai

Keterangan:

1. Nilai *Tho*
2. Nilai *Alpha*
3. Nilai *Beta*
4. Nilai Gama
5. Niali *Delta*
6. Back, untuk kembali ke halaman sebelumnya

4.4.5 Perancangan Antarmuka Halaman Query Tambahan

Pada laman ini digunakan untuk menampilkan tambahan query yang telah terurut dengan bobot terbesar dari tiap query aslinya, yaitu sebanyak 25 *query*, serta akan disediakan inputan *query* di dalamnya untuk dilakukan proses pencarian, yang ditunjukkan pada Gambar 4.12.



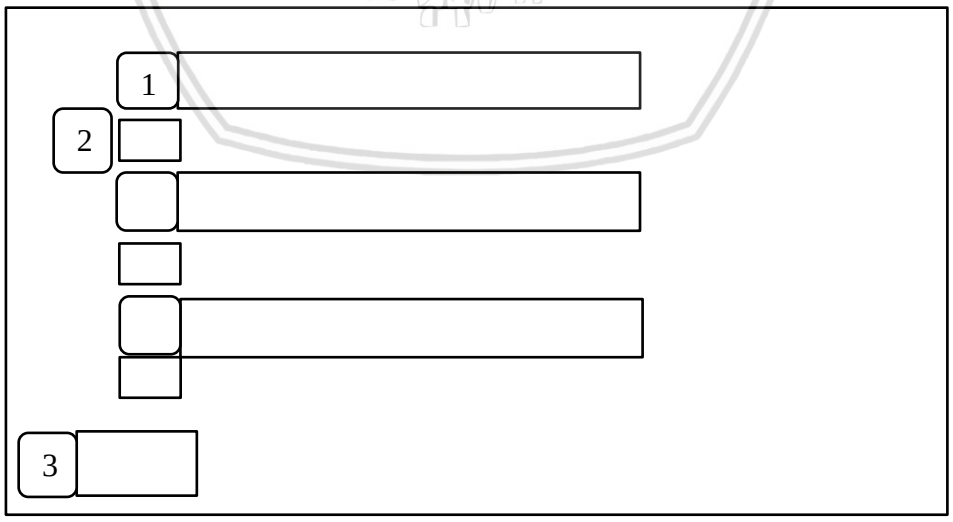
Gambar 4.12 Perancangan Antarmuka Halaman Query Tambahan

Keterangan:

1. Tombol untuk akses *query* tambahan dari 1-25 *query*
2. Tombol 'Back', untuk kembali ke halaman sebelumnya

4.4.6 Perancangan Antarmuka Halaman Hasil Pencarian Query Tambahan

Pada halaman ini berisi perangkingan dokumen hasil dari pencarian *query* yang diinputkan, beserta *checkbox* untuk tempat menilai dokumen sebagai dokumen relevan atau tidak. Berikut ditunjukkan pada Gambar 4.13.



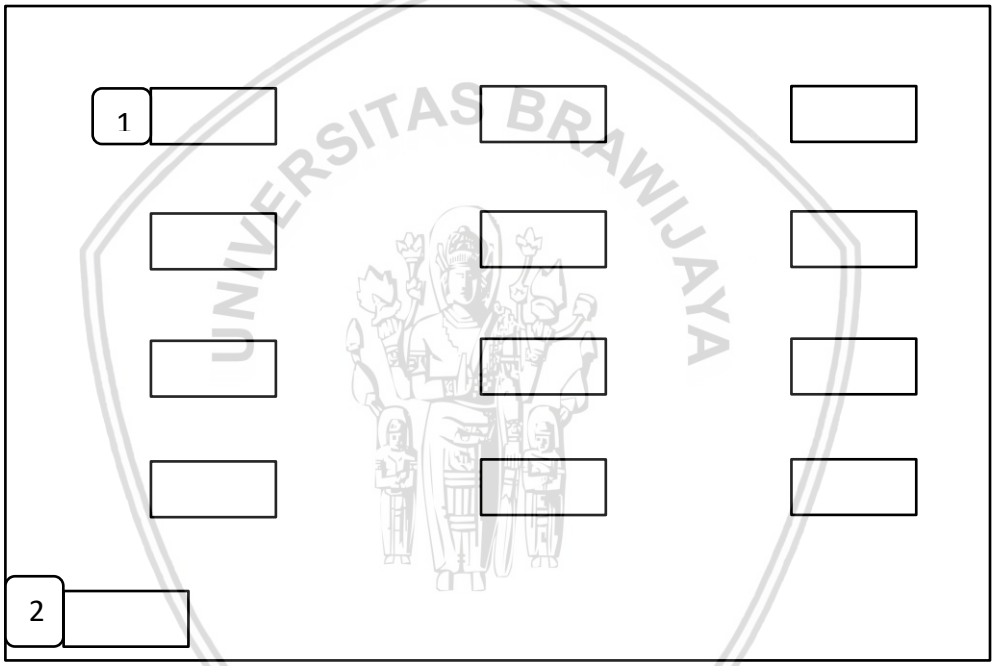
Gambar 4.13 Perancangan Antarmuka Halaman Hasil Pencarian Query Tambahan

Keterangan:

1. Perangkingan dokumen hasil dari pencarian *query*
2. *Checkbox* untuk menilai dokumen perangkingan hasil pencarian. Jika centang diberikan pada dokumen maka dokumen tersebut termasuk dokumen relevan dan muncul huruf 'R' disebelah *checkbox*
3. Tombol 'Back', untuk kembali ke halaman sebelumnya

4.4.7 Perancangan Antarmuka Halaman *Rocchio Relevance Feedback*

Pada laman ini digunakan untuk menampilkan tambahan query dengan metode *rocchio relevance feedback* dari query aslinya, serta akan disediakan inputan *query* di dalamnya untuk dilakukan proses pencarian, yang ditunjukkan pada Gambar 4.14.



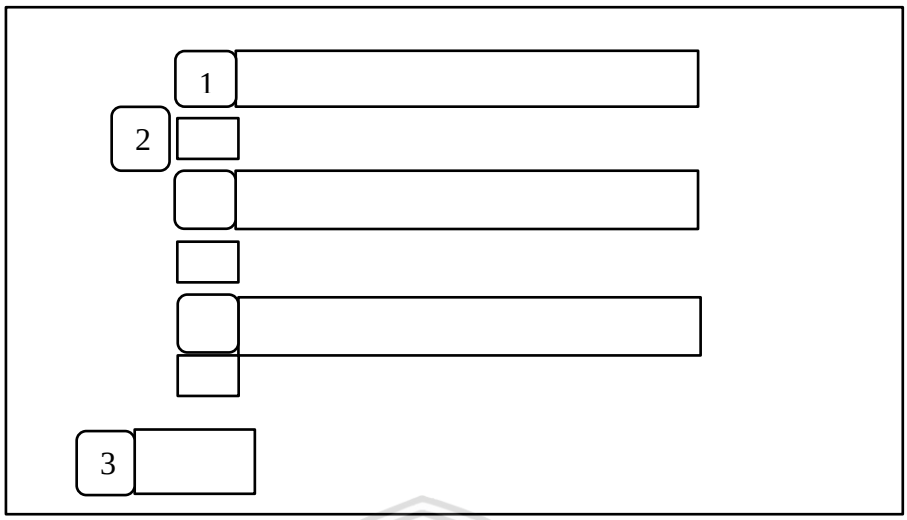
Gambar 4.14 Perancangan Antarmuka Halaman *Rocchio Relevance Feedback*

Keterangan:

1. Tombol untuk akses *query* tambahan dari 1-25 *query*
2. Tombol 'Back', untuk kembali ke halaman sebelumnya

4.4.8 Perancangan Antarmuka Halaman Hasil Pencarian *Rocchio Relevance Feedback*

Pada halaman ini berisi perangkingan dokumen hasil dari pencarian *query* yang diinputkan, beserta *checkbox* untuk tempat menilai dokumen sebagai dokumen relevan atau tidak. Berikut ditunjukkan pada Gambar 4.15.



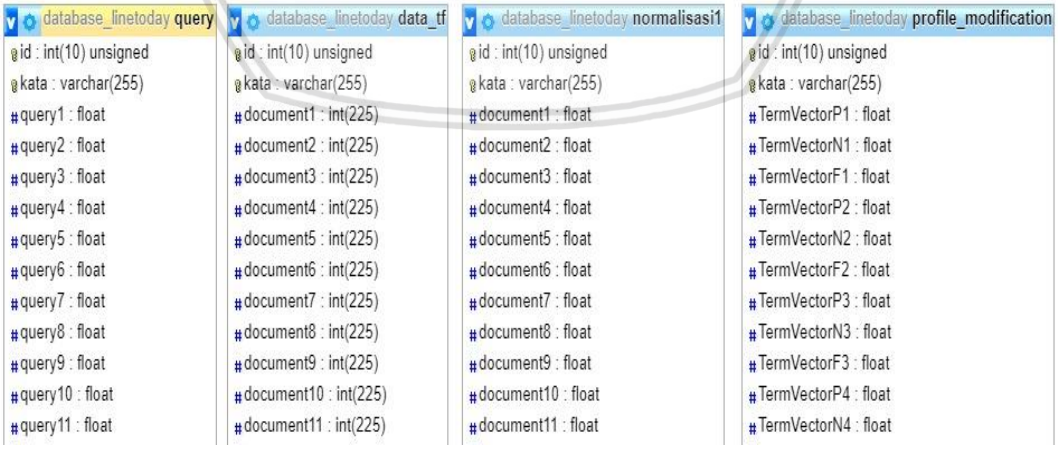
Gambar 4.15 Perancangan Antarmuka Halaman Hasil Pencarian *Rocchio Relevance Feedback*

Keterangan:

1. Perangkingan dokumen hasil dari pencarian *query*
2. *Checkbox* untuk menilai dokumen perangkingan hasil pencarian. Jika centang diberikan pada dokumen maka dokumen tersebut termasuk dokumen relevan dan muncul huruf 'R' disebelah *checkbox*
3. Tombol 'Back', untuk kembali ke halaman sebelumnya

4.5 Perancangan *Database*

Perancangan ini dibangun agar dapat menyimpan data dari informasi dari query, perhitungan, normalisasi, dan profile modification yang didapatkan. Perancangan tabel *database* yang dibangun ditunjukkan pada Gambar 4.16.



Gambar 4.16 Perancangan Tabel *Database*

4.5.1 Tabel *Query*

Pada tabel ini digunakan untuk penyimpanan informasi pada *query* yang ditunjukkan pada Tabel 4.23.

Tabel 4.23 *Database Query*

No	Nama Field	Type	Size
1	Id	int	10
2	kata	varchar	255
3	*Query1*	float	
4	*Query2*	float	
5	*Query3*	float	
6	*Query4*	float	
7	*Query5*	float	
8	*Query6*	float	
9	*Query7*	float	
10	*Query8*	float	

4.5.2 Tabel Data TF

Pada tabel ini dilakukan untuk menyimpan data pada *term frequency*. Berikut merupakan tabel data TF yang ditunjukkan pada Tabel 4.24.

Tabel 4.24 *Database* Data TF

No	Nama Field	Type	Size
1	id	int	10
2	kata	varchar	255
3	document1	int	255
4	document2	int	255
5	document3	int	255
6	document4	int	255
5	document5	int	255
6	document6	int	255
7	document7	int	255
8	document8	int	255
9	document9	int	255
10	document10	int	255

4.5.3 Tabel Normalisasi

Pada tabel ini dilakukan untuk menyimpan data yang berkaitan dengan normalisasi data. Berikut merupakan tabel normalisasi yang ditunjukkan pada Tabel 4.25.

Tabel 4.25 *Database* Normalisasi

No	Nama Field	Type	Size
1	id	int	10
2	kata	varchar	255
3	document1	float	
4	document2	float	
5	document3	float	

Tabel 4.25 *Database* Normalisasi (lanjutan)

No	Nama Field	Type	Size
6	document4	float	
5	document5	float	
6	document6	float	
7	document7	float	
8	document8	float	
9	document9	float	
10	document10	float	

4.5.4 Tabel *Profile Modification*

Pada tabel ini dilakukan untuk menyimpan data yang berkaitan dengan profile modifikasi, yang terdiri dari Term Vector P, N, maupun F. Berikut merupakan tabel profile modification yang ditunjukkan pada Tabel 4.26.

Tabel 4.26 *Database Profile Modification*

No	Nama Field	Type	Size
1	id	int	10
2	kata	varchar	255
3	TermVectorP1	float	
4	TermVectorN1	float	
5	TermVectorF1	float	
6	TermVectorP2	float	
5	TermVectorN2	float	
6	TermVectorF2	float	
7	TermVectorP3	float	
8	TermVectorN3	float	
9	TermVectorF3	float	
10	TermVectorP4	float	

4.6 Perancangan Pengujian dan Analisis

Pada perancangan pengujian ini dilakukan untuk mengukur tingkat kesalahan yang tersedia saat pengimplementasian metode *Extended Rocchio Relevance Feedback*. Pemanfaatan pada tabel *confusion matrix* digunakan untuk mempermudah dalam perhitunga *Recall, Precision, F-Measure,* dan nilai akurasinya. Pengujian akan dilakukan dengan 2 skenario pengujian dimana pada skenario pengujian pertama akan menguji variabel σ, α, β, γ, dan Δ (jika memenuhi syarat) di tiap *query*-nya, yang diuji *query* sebelum dimodifikasi dan *query* setelah dimodifikasi. Berikut merupakan hasil pengujian 1 yang dilakukan pada ke-lima variabel tersebut ditunjukkan pada Tabel 4.25.

Tabel 4.27 Skenario Pengujian 1

σ	α	β	γ	Δ	Indikator	Query		Kenaikan
						Awal	Baru	
					Precision			
					Recall			
					F-Measure			

Kemudian pada skenario pengujian kedua, dilakukan pengujian pada tiap *query* dengan mengukur nilai *precision, recall, f-measure,* dan *akurasi* pada banyak query tambahan dari *query* asli. Berikut merupakan perancangan tabel pengujian 2 yang ditunjukkan pada Tabel 4.26.

Tabel 4.28 Skenario Pengujian 2

Query	*Precision*	*Recall*	*F-Measure*	Akurasi
1 kata				
2 kata				

Pada skenario pengujian 3 ini dilakukan pengujian pada *threshold* peringkat *K* dengan *P@K,* dimana berguna untuk menghitung persentase dokumen relevan teratas sejumlah *K* yang sesuai dengan *query* dan mengabaikan dokumen yang berada di peringkat bawah *K.*

Tabel 4.29 Skenario Pengujian 3

No	Query	Peringkat	Ket.	P@K	Tambahan Kata
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

Skenario perancangan pengujian berikutnya dilakukan perbandingan antara metode *Extended Rocchio Relevance Feedback* dengan metode sebelumnya, yairu *Rocchio Relevance Feedback.* Indikator pengujian yang digunakan, ialah *precision, recall, f-measure,* dan akurasi.

Tabel 4.30 Skenario Pengujian 4

Metode	*Precision*	*Recall*	*F-Measure*	Akurasi
Extended Rocchio				
Rocchio				

4.7 Spesifikasi Sistem

Sistem yang dibangun digunakan untuk mempermudah pencarian dokumen, agar lebih spesifik dengan adanya tambahan *query* dari *query* asli atau sebelumnya yang kita inputkan. SIstem mengacu pada proses dan hasil analisis kebutuhan, serta perancangan yang dibahas pada bab sebelumnya. Spesifikasi sistem terdiri dari dua subbab, yaitu spesifikasi perangkat keras dan perangkat lunak.

4.7.1 Spesifikasi Perangkat Keras

Sistem yang dibangun merupakan *Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback*, yang memanfaatkan perangkat keras computer atau PC dengan spesifikasi seperti berikut ini, diantaranya:

a. Processor Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz 2.30 GHz
b. RAM 4.00GB

4.7.2 Spesifikasi Perangkat Lunak

Sistem yang dibangun merupakan *Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback*, yang memanfaatkan perangkat lunak computer atau PC dengan spesifikasi seperti berikut ini, diantaranya:

a. OS Windows 10 Professional 64 bit
b. XAMPP v3.2.1
c. Notepad++ Text Editor
d. Bahasa Pemrograman PHP dan HTML version : 5.6.3
e. Google Chrome Version 66.0.3359.181 (Official Build) (64-bit)
f. Materialize (Front-end framework)

4.8 Batasan Implementasi

Batasan implementasi ini berasal dari proses yang dilakukan sistem yang sesuai dengan perancangan yang dibangun. Batasan ini menunjukkan ruang lingkup dari sistem. Berikut beberapa batasan-batasan implementasi dari sistem *Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback*, diantaranya:

a. Sistem dirancangan dengan dan dapat diakses menggunakan applikasi berbasis web
b. Metode yang digunakan, ialah *Extended Rocchio Relevance Feedback*
c. Data latih yang digunakan berasal dari artikel berita sebanyak 200 data dari LINE TODAY, sedangkan data uji sebanyak 25 *query*
d. *Output* yang dihasilkan merupakan tambahan kata dari *query* sebelumnya atau *query* asli dan dilengkapi dengan hasil tamilan *query* yang diinputkan, yaitu berupa dokumen-dokumen berita yang berkaitan dengan *query* tersebut
e. Penentuan dokumen relevan maupun tidak relevan ditentukan oleh 3 orang mahasiswa

4.9 Implementasi

Sistem ini terdiri dari beberapa proses, dimulai dari *preprocessing,* kemudian *term weighting* atau TF.IDF, *Cosine Similarity,* dan metode *Extended Rocchio Relevance Feedback*

4.9.1 *Preprocessing*

Pada tahapan ini melalui beberapa proses di dalamnya, yaitu proses *cleansing*, untuk mengahapus komponen-komponen yang tidak penting, seperti tag html dan karakter lainnya selain huruf, *case folding,* untuk mengubah semua huruf menjadi *lowercase* atau huruf kecil, *tokenization,* untuk memisahkan setiap kata menjadi token, *filtering,* dilakukan penyaringan kata yang dianggap tidak penting, dan *stemming,* untuk mengubah tiap kata menjadi kata dasar.

4.9.1.1 *Cleansing*

Tahapan ini dilakukan untuk menghapus komponen tidak penting, seperti penghapusan pada tag HTML, URL, dan lain sebagainya. Penghapusan angka, tanda baca, dan karakter lainnya selain huruf juga dihilangkan. Berikut implementasi dari proses *Cleansing* ditunjukkan dalam *Source Code* 4.1.

No	*Source Code*
1	`$URL1 = "@(https?://([-\w\.]+[-\w])+(:\d+)?(/([\w/_\.#-]*(\?\S+)?[^\.\s])?)?)@";`
2	`  $Clean = preg_replace($URL1, ' ', $data_dokumen);`
3	`$URL2 = "{(([\w]+:)?//)?(([\d\w]|%[a-fA-f\d]{2,2})+(:([\d\w]|%[a-fA-f\d]{2,2})+)?@)?([\d\w][-\d\w]{0,253}[\d\w]\.)+[\w]{2,63}(:[\d]+)?(/([-+_~.\d\w]|%[a-fA-f\d]{2,2})*)*(\?(&?([-+_~.\d\w]|%[a-fA-f\d]{2,2})=?)*)?(#([-+_~.\d\w]|%[a-fA-f\d]{2,2})*)?}";`
4	`$Clean = preg_replace($URL2, ' ', $Clean);`
5	`$menghapus_Simbol ="/[^a-zA-Z]/";`
6	`$Clean = preg_replace($menghapus_Simbol, ' ', $Clean);`

Source Code 4.1 Implementasi *Cleansing*

Berikut merupakan penjelasan dari Implementasi *Cleansing* yang ditunjukkan pada Penjelasan *Source Code* 4.1.

No	Penjelasan
1	`$URL1` berisi url 'https'
2	Fungsi `preg_replace`, digunakan untuk mengganti karakter yang tidak diinginkan dengan cara diganti dengan sapasi atau *whitespace*
3	`$URL2` berisi karakter-karakter
4	Fungsi `preg_replace`, digunakan untuk mengganti karakter yang tidak diinginkan dengan cara diganti dengan sapasi atau *whitespace*
5	Menghapus simbol selain huruf a-z, huruf kecil maupun kapital
6	Fungsi `preg_replace`, digunakan untuk mengganti karakter yang tidak diinginkan dengan cara diganti dengan sapasi atau *whitespace*

Penjelasan Source Code 4.1 *Cleansing*

4.9.1.2 *Case Folding*

Pada proses ini dilakukan dengan mengubah semua kata menjadi huruf kecil atau *lowercase*. Berikut implementasinya ditunjukkan pada *Source Code* 4.2.

No	*Source Code*
1	`$CF = strtolower($Clean);`

Source Code 4.2 Implementasi *Case Folding*

Berikut merupakan penjelasan *Source Code* dari Implementasi *Case Folding* yang ditunjukkan pada Penjelasan *Source Code* 4.2.

No	Penjelasan
1	Fungsi ini untuk mengubah tiap kata menjadi huruf kecil

Penjelasan Source Code 4.2 *Case Folding*

4.9.1.3 *Tokenization*

Pada proses ini dilakukan dengan memisahkan tiap kata dengan *whitespace*. Berikut merupakan implementasinya ditunjukkan pada *Source Code* 4.3.

No	*Source Code*
1	`$Tokenization = explode(' ', $CF);`
2	`$Tokenization = array_filter($Tokenization);`
3	`sort($Tokenization);`

Source Code 4.3 Implementasi *Tokenization*

Berikut ini merupakan penjelasan *Source Code* dari Implementasi *Tokenization* yang ditunjukkan pada Penjelasan *Source Code* 4.3.

No	Penjelasan
1	Fungsi untuk memisahkan string yang dipecah dengan spasi
2	Untuk menghilangkan nilai kosong

Penjelasan Source Code 4.3 *Tokenization*

4.9.1.4 *Filtering*

Pada proses ini dilakukan penghapusan kata tidak penting dan melihat list dari *stopword*. Berikut merupakan implementasinya pada *Source Code* 4.4.

No	*Source Code*
1	`$data_stopword_list_file = "0-0-stopword_list.txt";`
2	`$read = fopen($data_stopword_list_file, "r");`
3	`$data_stopword_list = fread($read, filesize($data_stopword_list_file));`
4	`$daftar_stopword_list = explode("\n", $data_stopword_list);`
5	`fclose($read);`
6	`$daftar_stopword_list = array_filter($daftar_stopword_list);`
	`$Filter = array_values(array_diff(array_map("trim", $Tokenization), array_map("trim", $daftar_stopword_list)));`

Source Code 4.4 Impelementasi *Filtering*

Berikut ini merupakan penjelasan *Source Code* dari Implementasi *Filtering* yang ditunjukkan pada Penjelasan *Source Code* 4.4.

No	Penjelasan
1	Data stopword list ada pada file "0-0-stopword_list.txt"
2-3	Untuk membuka dan membaca list kata di stopword
4	Memisahkan tiap kata string dengan *"enter"*
5	Untuk menutup file yang terbaca
6	Merupakan proses filtering. Dilakukan penghapusan pada nilai kosong pada nilai kosong dalam list stopword. array_dif(), untuk hapus array dengan memisahkan kata yang ada di tokenisasi dan tersedia di list *stopword*

Penjelasan Source Code 4.4 *Filtering*

4.9.1.5 *Stemming*

Pada proses ini dilakukan dengan mengubah semua kata hasil dari *filtering* menjadi kata dasar. Berikut merupakan implementasinya ditunjukkan pada *Source Code* 4.5.

No	*Source Code*
1	`$Stemming = $Filter;`
2	`foreach ($Stemming as &$daftar_kata_stemming){`
3	`    $kata1 = Del_Inflection_Suffixes($daftar_kata_stemming);`
4	`    $kata2 = Del_Derivation_Suffixes($kata1);`
5	`    $kata3 = Del_Derivation_Prefix($kata1);`
6	`    $daftar_kata_stemming = $kata3;`
	`    continue;`
	`  }`

Source Code 4.5 Implementasi *Stemming*

Berikut ini merupakan penjelasan *Source Code* dari Implementasi *Stemming* yang ditunjukkan pada Penjelasan *Source Code* 4.5.

No	Penjelasan
1-2	Untuk mengambil data pada filtering dan dilakukan perulangan atau pengecekan tiap kata
3-5	Dilakukan proses penghapusan inflection suffixes, derivation suffixes, dan penghapusan derivation prefix
6	List kata stemming yang digunakan ada pada proses terakhir, yaitu setelah penghapusan derivation prefix

Penjelasan Source Code 4.5 *Stemming*

4.9.2 *Term Weighting* (TF.IDF)

Pembobotan dilakukan menggunakan TF.IDF yang digunakan untuk mengidentifikasi keunikan atau kemunculan kata yang berbeda tiap dokumen. Dalam perhitungannya, digunakan pada persamaan (2.1) sampai (2.3) dan persamaan (2.4), yaitu sampai perhitungan $W_{t,d}$ yang dilakukan normalisasi. Berikut merupakan implementasi *Term Weighting* ditunjukkan pada *Source Code* 4.6.

No	*Source Code*
	//Proses Pembobotan Nilai data_tf
1	$bobot_nilai_TF =0;
2	$hitung_pembobotan = NULL;
3	foreach ($nilai_TF as $key => $value) {
4	$bobot_nilai_TF = 1+log10($value);
5	$hitung_pembobotan[$key]=$bobot_nilai_TF;
	}
	//Mengambil Nilai IDF
6	foreach ($hitung_pembobotan as $key => $value) {
7	$daftar_kata = $key;
8	$nilai_IDF = "SELECT idf1 FROM data_tf WHERE kata='$daftar_kata'";
9	$ambil_IDF = mysqli_query($mysqli, $nilai_IDF);
10	$row1=mysqli_fetch_array($ambil_IDF);
11	$nilai = $row1[0];
12	if($nilai != 0){
13	$idf[$daftar_kata] = $nilai;
	}
	}
14	if ($idf==NULL){
15	header("Location:1-4-Query.php?status=Query tidak ditemukan di dalam Database!");
	exit;
	}
	//Perhitungan Perkalian TF.IDF
16	$hitung_tf_idf = NULL;
17	foreach ($idf as $key => $value) {
18	foreach ($hitung_pembobotan as $key1 => $value1) {
19	if ($key == $key1){
20	$hitung_tf_idf[$key1] = $value*$value1;
	}
	}
	}
	//Menghitung Nilai Akar Pangkat tiap Dokumen
21	$jumlah = 0;
22	$hitung_pangkat = 0;
23	foreach ($hitung_tf_idf as $key => $value) {
24	$hitung_pangkat = pow($value,2);
25	$jumlah = $jumlah + $hitung_pangkat;
	}
26	$jumlah = sqrt($jumlah);
	//Menghitung Normalisasi
27	$normalisasi = NULL;
28	foreach ($hitung_tf_idf as $key => $value) {
29	$hasil_normalisasi = $value/$jumlah;
30	$normalisasi[$key]=$hasil_normalisasi;
	}

Source Code 4.6 Implementasi *Term Weighting*

Berikut ini merupakan penjelasan *Source Code* dari Implementasi *Term Weighting* yang ditunjukkan pada Penjelasan *Source Code* 4.6.

No	Penjelasan
1-5	Untuk melakukan pembobotan nilai data tf dengan menggunakan rumus
6-13	Untuk ambil nilai idf pada daftar kata di database
14-15	Jika nilai idf bernilai kosong maka akan muncul tampilan "Query tidak ditemukan di dalam Database!"
16-20	Untuk proses perhitungan TF.IDF
21-26	Dilakukan proses perhitungan akar pangkat di tiap dokumen
27-30	Proses perhitungan normalisasi

Penjelasan Source Code 4.6 *Term Weighting*

4.9.3 *Cosine Similarity*

Pada proses inidilakukan pada data latih terhadap data uji yang dipakai guna mengetahui nilai kemiripan atau kesesuaian. Data latih yang digunakan sebanyak 200 data dengan 25 data uji. Perhitungan dilakukan dengan persamaan (2.6). Hasil dari tampilan dokumen pada proses ini hanya dokumen dengan nilai *cosine similarity* bernilai lebih besar dari 0. Berikut merupakan implementasinya ditunjukkan pada *Source Code* 4.7.

No	*Source Code*
1	`for ($i = 1; $i <= 200; $i++){`
2	`  $hasil = 0;`
3	`  $document = "document".$i;`
4	`  foreach ($perhitungan_cossim as $key => $value) {`
5	`    $nilai_normalisasi = "SELECT $document FROM normalisasi1 WHERE kata='$key'";`
6	`    $ambil_nilai = mysqli_query($mysqli, $nilai_normalisasi);`
7	`    $row1 = mysqli_fetch_array($ambil_nilai);`
8	`    $nilai = $row1[0];`
9	`    if($nilai != 0){`
10	`      $cossim = $nilai * $value;`
11	`      $hasil = $hasil + $cossim;`
	`    }`
12	`    else{`
13	`      $cossim = 0;`
14	`      $hasil = $hasil + $cossim;`
	`    }`
	`  }`
15	`  if($hasil != 0){`
16	`    $array_document[$document] = $hasil;`
17	`    $angka_data[$document] = $i-1;`
	`  }`
	`}`
18	`arsort($array_document);`

Source Code 4.7 Implementasi *Cosine Similarity*

Berikut ini merupakan penjelasan *Source Code* dari Implementasi *Cosine Similarity* yang ditunjukkan pada Penjelasan *Source Code* 4.7.

No	Penjelasan
1-7	Dilakukan pengecekan tiap kata dan ambil data dari database, yaitu dari data normalisasi
8-18	Proses perhitungan *cosine similarity*

Penjelasan Source Code 4.7 *Cosine Similarity*

4.9.4 *Extended Rocchio Relevance Feedback*

4.9.4.1 *Relevance Feedback*

a. *Average Weight Term Vector*

Proses ini mendapatkan nilai rata-rata dari hasil normalisasi sebelumnya, dimana *Average Weight Term Vector* P, N, dan F didapat dengan mengacu pada dokumen relevan untuk P, tidak relevan untuk N, dan F untuk dokumen dengan nilai *cosine similarity* 0. Implementasi ditunjukkan pada *Source Code* 4.8.

No	*Source Code*
1	`  $i = 0;`
2	`  $AvgTermVectorP = NULL;`
3	`  foreach ($Relevan1 as $keyR => $valueR) {`
4	`    $i = $i+1;`
5	`    $doc_Relevan = "doc_Relevan".$i;`
6	`    $tf_R = NULL;`
7	`    $document = "document".$valueR;`
	`    //Ambil Daftar data_tf`
8	`    $ambil_document_document = "SELECT kata FROM data_tf";`
9	`    $select_document = mysqli_query($mysqli, $ambil_document_document);`
10	`    echo $mysqli->error;`
11	`    while($row = mysqli_fetch_array($select_document)){`
12	`      $daftar_kata = $row[0];`
13	`      $ambil_document_dok_TF = "SELECT $document FROM normalisasi1 WHERE kata = '$daftar_kata'";`
14	`      $select_dok_TF = mysqli_query($mysqli, $ambil_document_dok_TF);`
15	`      $row1=mysqli_fetch_array($select_dok_TF);`
16	`      $nilai = $row1[0];`
17	`      if($nilai != 0){`
18	`        $tf_R[$daftar_kata] = $nilai;`
	`      }`
	`    }`
19	`    $$doc_Relevan = $tf_R;`
	`  }`
20	`  $k = 0;`
21	`  $AvgTermVectorN = NULL;`
22	`  foreach ($Tidak_Relevan1 as $keyR => $valueR) {`
23	`    $k = $k+1;`
24	`    $doc_TidakRelevan = "doc_TidakRelevan".$k;`
25	`    $tf_TR = NULL;`

Source Code 4.8 Implementasi *Average Weight Term Vector*

No	*Source Code*
26	`$document = "document".$valueR;`
	`//Ambil Daftar data_tf`
27	`$ambil_document_document = "SELECT kata FROM data_tf";`
28	`$select_document = mysqli_query($mysqli, $ambil_document_document);`
29	`echo $mysqli->error;`
30	`while($row = mysqli_fetch_array($select_document)){`
31	`$daftar_kata = $row[0];`
32	`$ambil_document_dok_TF = "SELECT $document FROM normalisasi1 WHERE kata = '$daftar_kata'";`
33	`$select_dok_TF = mysqli_query($mysqli, $ambil_document_dok_TF);`
34	`$row1=mysqli_fetch_array($select_dok_TF);`
35	`$nilai = $row1[0];`
36	`if($nilai != 0){`
37	`$tf_TR[$daftar_kata] = $nilai;`
	`}`
	`}`
38	`$$doc_TidakRelevan = $tf_TR;`
	`}`
39	`$m = 0;`
40	`$AvgTermVectorF = NULL;`
41	`foreach ($Dokumen1 as $keyR => $valueR) {`
42	`$m = $m+1;`
43	`$doc_0 = "doc_0".$m;`
44	`$tf_0 = NULL;`
45	`$document = "document".$valueR;`
	`//Ambil Daftar data_tf`
46	`$ambil_document_document = "SELECT kata FROM data_tf";`
47	`$select_document = mysqli_query($mysqli, $ambil_document_document);`
48	`echo $mysqli->error;`
49	`while($row = mysqli_fetch_array($select_document)){`
50	`$daftar_kata = $row[0];`
51	`$ambil_document_dok_TF = "SELECT $document FROM normalisasi1 WHERE kata = '$daftar_kata'";`
52	`$select_dok_TF = mysqli_query($mysqli, $ambil_document_dok_TF);`
53	`$row1=mysqli_fetch_array($select_dok_TF);`
54	`$nilai = $row1[0];`
55	`if($nilai != 0){`
56	`$tf_0[$daftar_kata] = $nilai;`
	`}`
	`}`
57	`$$doc_0 = $tf_0;`
	`}`
	`//Ambil Daftar data_tf`
58	`$ambil_document_document = "SELECT kata FROM data_tf";`
59	`$ambil_document_document = mysqli_query($mysqli, $ambil_document_document);`

Source Code 4.8 Implementasi *Average Weight Term Vector* (lanjutan)

No	Source Code
60	`echo $mysqli->error;`
61	`while($row = mysqli_fetch_array($ambil_document_document)){`
62	`$daftar_kata = $row[0];`
63	`$AvgTermVectorP[$daftar_kata] = 0;`
64	`$AvgTermVectorN[$daftar_kata] = 0;`
65	`$AvgTermVectorF[$daftar_kata] = 0;`
	`}`
66	`$i = $i;`
67	`$TotalJumlahRelevan = 0;`
68	`for ($j = 1; $j <= $i ; $j++){`
69	`$docR = "doc_Relevan".$j;`
70	`foreach ($$docR as $key => $value) {`
71	`if ($value == 0){`
72	`if($AvgTermVectorP[$key]==0){`
73	`$AvgTermVectorP[$key]==0;`
	`}`
	`}`
74	`else{`
75	`$TotalJumlahRelevan = $TotalJumlahRelevan + 1;`
76	`if($AvgTermVectorP[$key]==0){`
77	`$AvgTermVectorP[$key]=($value/$TotalJumlahRelevan);`
	`}`
78	`else{`
79	`$AvgTermVectorP[$key]=(($AvgTermVectorP[$key]+$value)/$TotalJumlahRelevan);`
	`}`
	`}`
	`}`
	`}`
80	`$k = $k;`
81	`$TotalJumlahTidakRelevan = 0;`
82	`for ($l = 1; $l <= $k ; $l++){`
83	`$docTR = "doc_TidakRelevan".$l;`
84	`foreach ($$docTR as $key => $value) {`
85	`if ($value == 0){`
86	`if($AvgTermVectorN[$key]==0){`
87	`$AvgTermVectorN[$key]==0;`
	`}`
	`}`
88	`else{`
89	`$TotalJumlahTidakRelevan = $TotalJumlahTidakRelevan + 1;`
90	`if($AvgTermVectorN[$key]==0){`
91	`$AvgTermVectorN[$key]=($value/$TotalJumlahTidakRelevan);`
	`}`
92	`else{`

Source Code 4.8 Implementasi *Average Weight Term Vector* (lanjutan)

No	*Source Code*

93	`$AvgTermVectorN[$key]=(($AvgTermVectorN[$key]+$value)/$Total JumlahTidakRelevan);`
	`}`
	`}`
	`}`
	`}`

94	`$m = $m;`
95	`$TotalJumlah0 = 0;`
96	`for ($n = 1; $n <= $m ; $n++){`
97	`$doc0 = "doc_0".$n;`
98	`foreach ($$doc0 as $key => $value) {`
99	`if ($value == 0){`
100	`if($AvgTermVectorF[$key]==0){`
101	`$AvgTermVectorF[$key]==0;` `}`
	`}`
102	`else{`
103	`$TotalJumlah0 = $TotalJumlah0 + 1;`
104	`if($AvgTermVectorF[$key]==0){`
105	`$AvgTermVectorF[$key]=($value/$TotalJumlah0);` `}`
106	`else{`

107	`$AvgTermVectorF[$key]=(($AvgTermVectorF[$key]+$value)/$Total Jumlah0);`
	`}`
	`}`
	`}`
	`}`

Source Code 4.8 Implementasi *Average Weight Term Vector* (lanjutan)

Berikut ini merupakan penjelasan *Source Code* dari implementasi *Average Weight Term Vector* yang ditunjukkan pada Penjelasan *Source Code* 4.8.

No	Penjelasan
1-19	Dilakukan proses pemanggilan data untuk perhitungan *Average Term Vector P,* dimana data diambil berdasar data value normalisasi pada proses pengambilan dokumen pada data tf yang diambil pada database
20-38	Dilakukan proses pemanggilan data untuk perhitungan *Average Term Vector N,* dimana data diambil berdasar data value normalisasi pada proses pengambilan dokumen pada data tf yang diambil pada database
39-57	Dilakukan proses pemanggilan data untuk perhitungan *Average Term Vector F,* dimana data diambil berdasar data value normalisasi pada proses pengambilan dokumen pada data tf yang diambil pada database
58-65	Mengambil data dari data tf dan kata yang digunakan untuk *term vector P, N,* dan *F* dalam database
66-79	Proses perhitungan *Average Term Vector P*
80-93	Proses perhitungan *Average Term Vector N*
94-107	Proses perhitungan *Average Term Vector F*

Penjelasan Source Code 4.8 *Average Weight Term Vector*

b. ***Term Vector***

Pada proses ini dilakukan untuk menentukan nilai *Term Vector P, N,* dan *F*. Untuk menentukan dokumen mana saja yang digunakan untuk *Term Vector* P, N, dan F dinilai berdasar dokumen relevan untuk P, dokumen tidak relevan untuk N, dan dokumen dengan nilai *cosine similarity* bernilai 0 untuk F. Implementasi ditunjukkan pada *Source Code* 4.9.

No	***Source Code***
	`//Menghitung Term Vector P, N, F`
	`//Menghitung Term Vector P`
1	`  foreach ($Query_Uji1 as $key => $value) {`
2	`    foreach ($AvgTermVectorP as $key1 => $value1) {`
3	`      if($value == $key1){`
4	`        $AvgTermVectorP[$key1] = 0;`
	`      }`
	`    }`
	`  }`
	`//Menghitung Term Vector N`
5	`  foreach ($Query_Uji1 as $key => $value) {`
6	`    foreach ($AvgTermVectorN as $key1 => $value1) {`
7	`      if($value == $key1){`
8	`        $AvgTermVectorN[$key1] = 0;`
	`      }`
	`    }`
	`  }`
	`//Menghitung Term Vector F`
9	`  foreach ($Query_Uji1 as $key => $value) {`
10	`    foreach ($AvgTermVectorF as $key1 => $value1) {`
11	`      if($value == $key1){`
12	`        $AvgTermVectorF[$key1] = 0;`
	`      }`
	`    }`
	`  }`
13	`  foreach ($AvgTermVectorP as $key => $value) {`
14	`    if($value != 0){`
15	`      $AvgTermVectorF[$key] = 0;`
	`    }`
	`  }`
16	`  foreach ($AvgTermVectorN as $key => $value) {`
17	`    if($value != 0){`
18	`      $AvgTermVectorF[$key] = 0;`
	`    }`
	`  }`
19	`$TermVectorP = $AvgTermVectorP;`
20	`$TermVectorN = $AvgTermVectorN;`
21	`$TermVectorF = $AvgTermVectorF;`

Source Code 4.9 Implementasi *Term Vector*

Berikut ini merupakan penjelasan *Source Code* dari implementasi *Term Vector* yang ditunjukkan pada Penjelasan *Source Code* 4.9.

No	Penjelasan
1-4	Proses untuk perhitungan *Term Vector P*
5-8	Proses untuk perhitungan *Term Vector N*
9-12	Proses untuk perhitungan *Term Vector F*
13-15	Dilakukan pengecekan pada *term vector P*, dimana akan bernilai 0 jika ada pada query asli dan terletak pada dok relevan
16-18	Dilakukan pengecekan pada *term vector N*, dimana akan bernilai 0 jika ada pada query asli dan terletak pada dok tidak relevan

Penjelasan Source Code 4.9 *Term Vector*

4.9.4.2 *Profile Modification*

Pada proses perhitungan ini dilakukan dengan persamaan (2.9) atau (2.10), dimana sebelumnya set nilai parameter dari σ, α, β, γ, dan Δ untuk digunakan dalam perhitungan. Pengujian dilakukan dengan set kelima nilai parameter dengan satu *query* 'Avenger Infinity War' yang akan ditemukan nilai-nilai parameter yang cocok untuk dipakai pada pengujian di tiap *query*. Berikut merupakan hasil implementasi dari salah satu *query* 'kisah unik' setelah ditentukan nilai-nilai parameternya, dimana nilai parameter yang terpilh dan yang digunakan untuk keseluruhan *query*, yang ditunjukkan pada *Source Code* 4.10.

No	*Source Code*
1	` <h6 style="font-style: italic;"><b>σ = 0.25 // α = 1.25 // β = 0.79 // γ = 0.28 // Δ = 0.54</b></h6>`
	`<?php`
	`  //PENGUJIAN 3`
2	`  $tho = 0.25;`
3	`  $alfa = 1.25;`
4	`  $beta = 0.79;`
5	`  $gama = 0.28;`
6	`  $delta = 0.54;`
7	`  foreach ($Hasil as $key => $value) {`
8	`    if ($V > $tho){`
9	`      $Hasil[$key] = (($alfa * $Qmod[$key]) + ($beta * $TermVectorP[$key]) - ($gama * $TermVectorN[$key]) + ($delta * $TermVectorF[$key]));`
10	`      $a[$key] = "(V = ".$V.") > (σ = ".$tho.")";`
11	`      $b[$key] = "V = α*Q_mod+β*P-γ*N+Δ*F";`
12	`      $c[$key] = "V = ((".$alfa." * ".$Qmod[$key].") + (".$beta." * ".$TermVectorP[$key].") - (".$gama." * ".$TermVectorN[$key].") + (".$delta." * ".$TermVectorF[$key]."))";`
	`    }`
13	`    else{`
14	`      $Hasil[$key] = (($alfa * $Q[$key]) + ($beta * $TermVectorP[$key]) - ($gama * $TermVectorN[$key]));`
15	`      $a[$key] = "(V = ".$V.") < (σ = ".$tho.")";`
16	`      $b[$key] = "V_new = α*Q+β*P-γ*N";`

Source Code 4.10 Implementasi *Profile Modification*

No	*Source Code*

17	`$c[$key] = "V_new = ((".$alfa." * ".$Qmod[$key].") + (".$beta." * ".$TermVectorP[$key].") - (".$gama." * ".$TermVectorN[$key]."))";`
	`}`
	`}`

Source Code 4.10 Implementasi *Profile Modification* (lanjutan)

Berikut ini merupakan penjelasan *Source Code* dari implementasi *Profile Modification* yang ditunjukkan pada Penjelasan *Source Code* 4.10.

No	Penjelasan
1-6	Inisialisasi parameter
8-12	Dilakukan kondsi perhitungan jika V>σ, maka akan dimasukkan ke dalam rumus `α*Q_mod+β*P-γ*N+Δ*F`
13-17	Dilakukan kondsi lainnya perhitungan jika V<σ, maka akan dimasukkan ke dalam rumus `α*Q+β*P-γ*N`

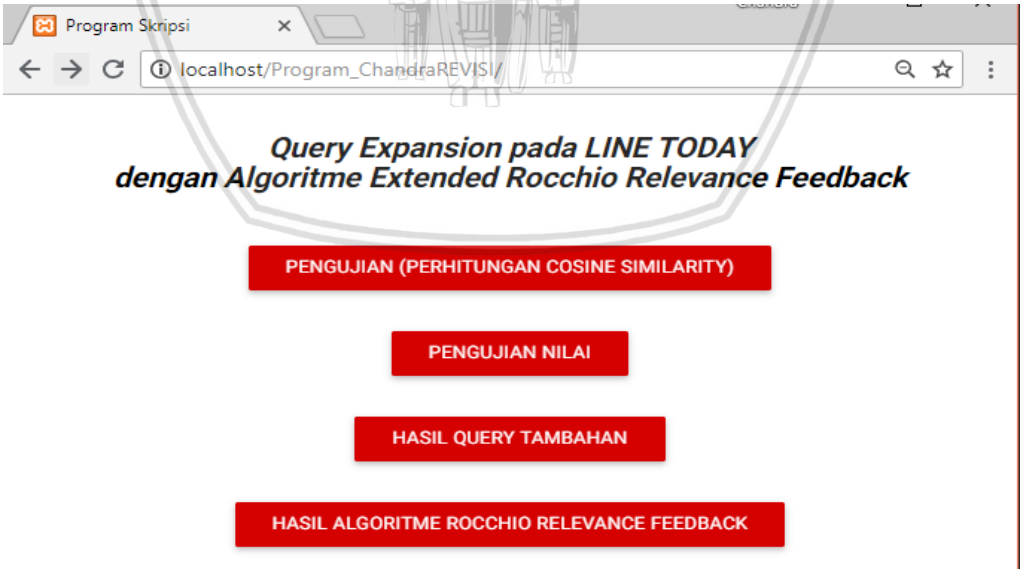
Penjelasan Source Code 4.10 *Profile Modification*

4.10 Implementasi Antar Muka

Antarmuka pada sistem ini dibangun untuk mempermudah pengguna dalam menjalankan program.

4.10.1 Tampilan Halaman Awal

Tampilan halaman awal ini terdiri dari 4 menu pilihan, yaitu pengujian untuk *query* asli atau saat proses *cosine similarity*, pengujian nilai parameter, *query* tambahan, dan menu hasil algoritme *rocchio relevance feedback*. Berikut halaman awal ditunjukkan pada Gambar 4.17.

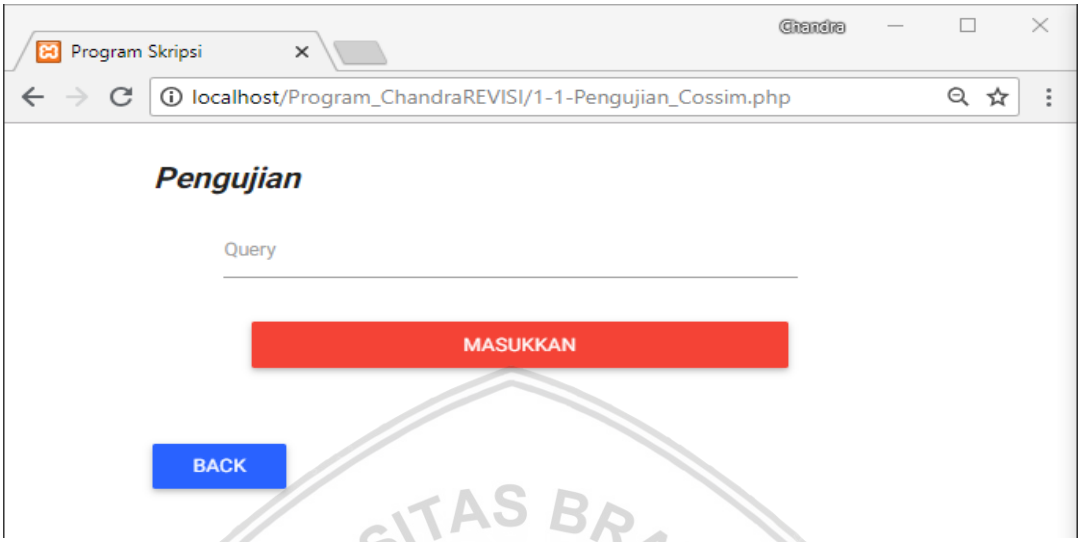


Gambar 4.17 Tampilan Halaman Awal

4.10.2 Tampilan Halaman Pengujian *Cosine Similarity*

Pada halaman ini disediakan *search box* untuk mengetikkan query yang akan diinputkan. Pengujian ini dilakukan dengan memanfaatkan perhitungan dari *cosine*

similarity dalam hasil perangkingannya. *Query* yang diinputkan bergantung dari apa yang diinputkan pengguna. Berikut halaman pengujian *query* ditunjukkan pada Gambar 4.18.



Gambar 4.18 Tampilan Halaman Pengujian *Cosine Similarity*

4.10.2.1 Tampilan Halaman Hasil Pencarian *Cosine Similarity*

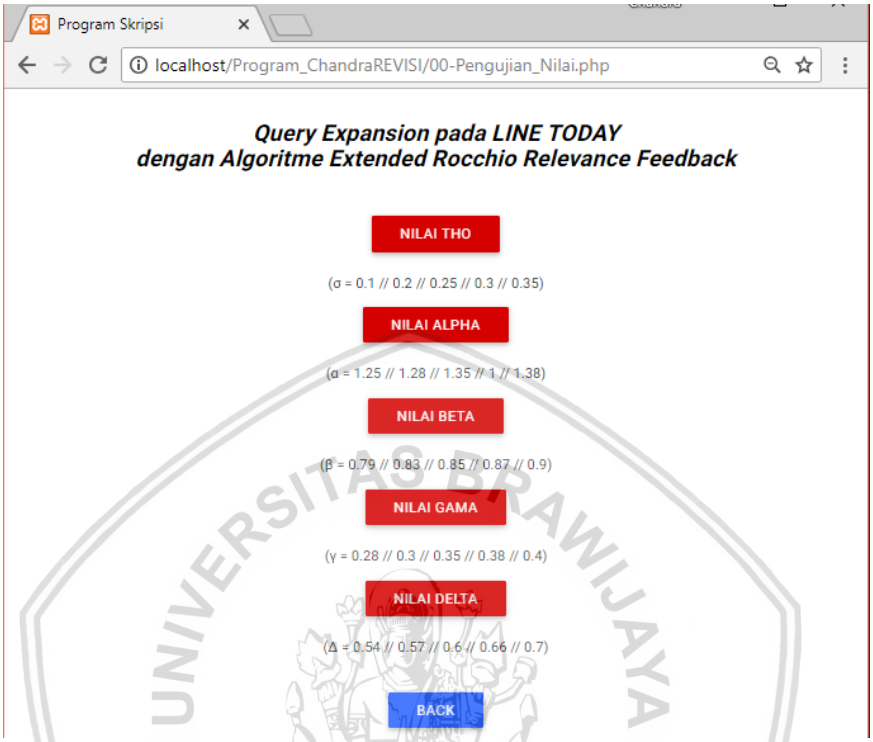
Halaman ini menampilkan dokumen-dokumen perangkingan dari hasil *query* yang telah diinputkan sebelumnya. Tiap dokumen akan memiliki *checkbox* untuk penilaian yang diberikan oleh pengguna, dimana jika *checkbox* tersebut dicentang, maka menandakan dokumen tersebut relevan. Berikut halaman hasil *query* yang diinputkan ditunjukkan Gambar 4.19.

Gambar 4.19 Tampilan Halaman Hasil Pencarian *Cosine Similarity*

4.10.3 Tampilan Halaman Pengujian Nilai

Pada halaman ini menampilkan nilai parameter yang akan diuji dengan menggunakan salah satu query. Salah satu *query* yang diuji, ialah “Avenger Infinity

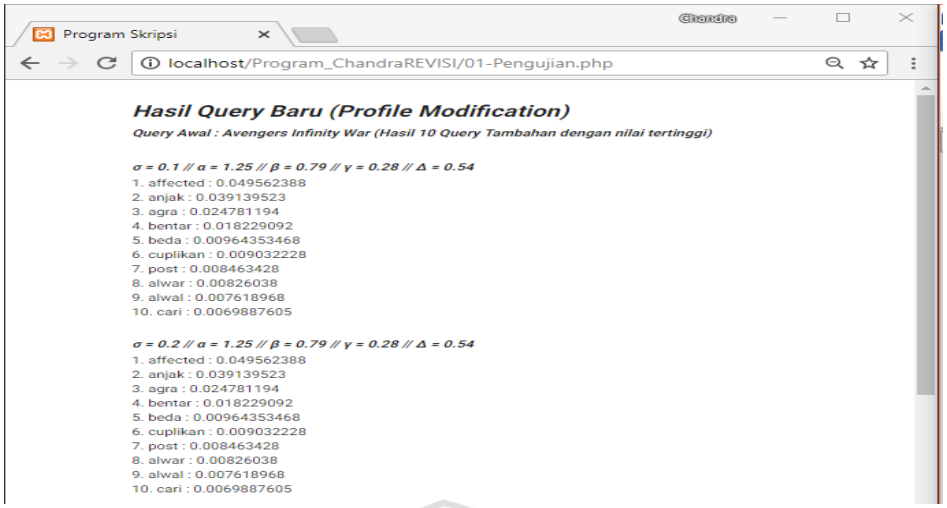
War", dimana parameter yang diuji, ialah parameter *tho, alpha, beta, gama,* dan *delta.* Parameter yang yang ada di tiap menu menujukkan pengujian untuk satu jenis parameter saja, yaitu diset dengan angka berbeda. Berikut merupakan tampilan halaman Pengujian Nilai yang ditunjukkan pada Gambar 4.20.



Gambar 4.20 Tampilan Halaman Pengujian Nilai

4.10.3.1 Tampilan Halaman Hasil Pengujian Nilai *Tho*

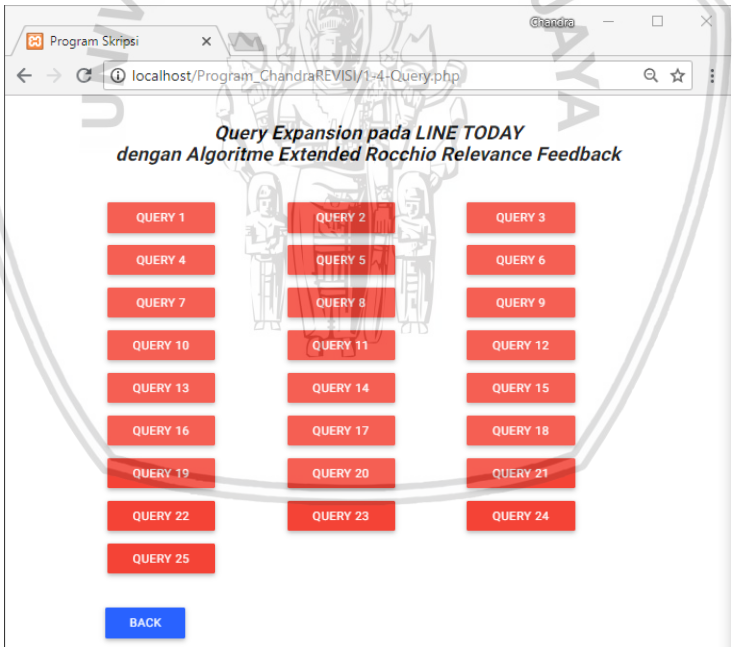
Pada halaman ini menunjukkan pengujian pada tiap nilai parameter. Jika menguji nilai tho (σ), maka nilai parameter lainnya akan disamakan untuk tiap nilai parameternya, sedangkan nilai σ diset dengan nilai yang berbeda. Setelah terpilih nilai σ yang dianggap memiliki nilai kenaikan tertinggi pada *precision, recall,* dan *f-measure,* maka dilanjutkan menguji nilai alpha (α) dengan nilai yang berbeda dan parameter lainnya diberi nilai yang sama, begitu juga nilai σ, karena sebelumnya nilai σ telah terpilih. Berikut merupakan halaman hasil pengujian pada nilai tho (σ) yang ditunjukkan pada Gambar 4.21.



Gambar 4.21 Tampilan Halaman Hasil Pengujian Nilai *Tho*

4.10.4 Tampilan Halaman Query Tambahan

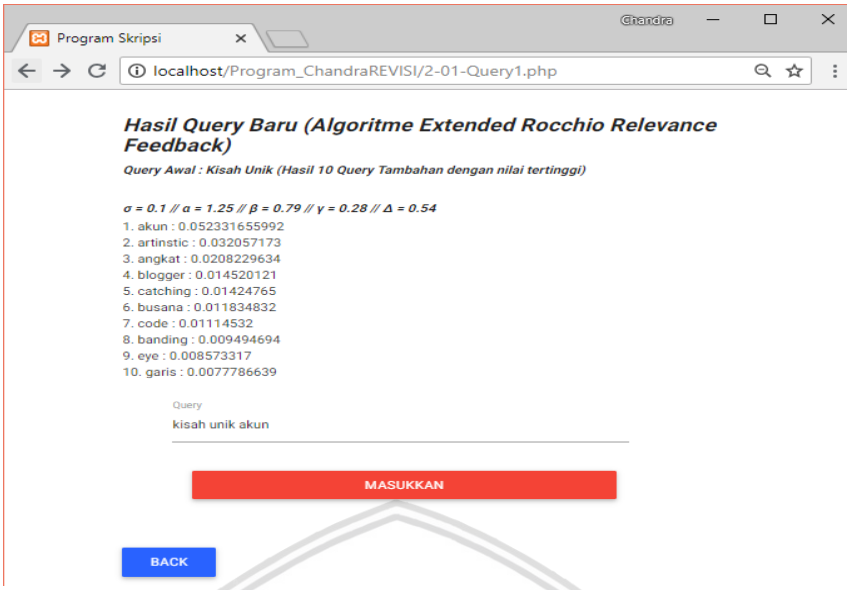
Pada halaman ini menampilkan *query* tambahan dari tiap query asli, dimana terdapat 25 data uji atau *query*. Berikut merupakan tampilan halaman *query* tambahan ditunjukkan pada Gambar 4.22.



Gambar 4.22 Tampilan Halaman *Query* Tambahan

4.10.4.1 Tampilan Halaman Hasil *Query* Tambahan

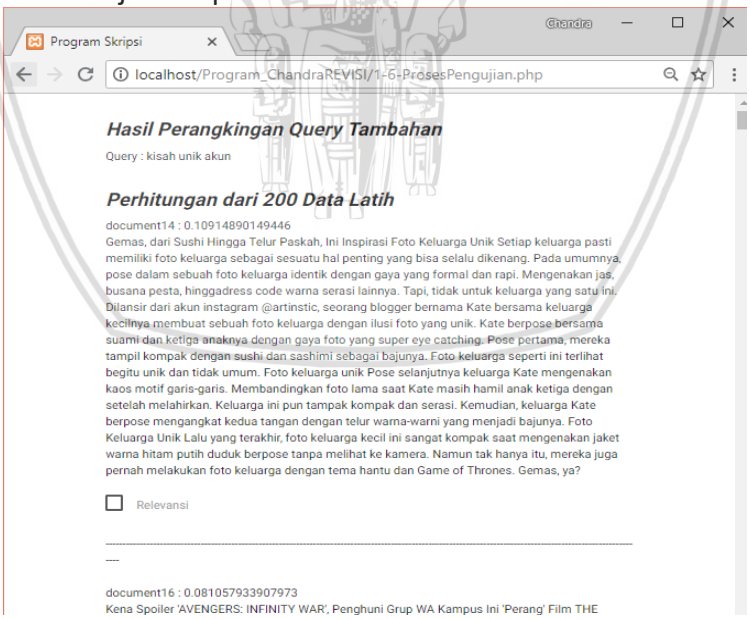
Halaman hasil *query* tambahan ini terdapat list *query* tambahan yang telah diranking dari nilai tertinggi hingga terendah, yaitu sebanyak 10 *query*. Dalam halaman tersebut juga tersedia *search box* untuk mengetikkan *query* yang ingin diinputkan. Berikut merupakan tampilan halaman hasil *query* tambahan yang ditunjukkan pada Gambar 4.23.



Gambar 4.23 Tampilan Halaman Hasil *Query* Tambahan

4.10.4.2 Tampilan Halaman Hasil Pencarian *Query* Tambahan

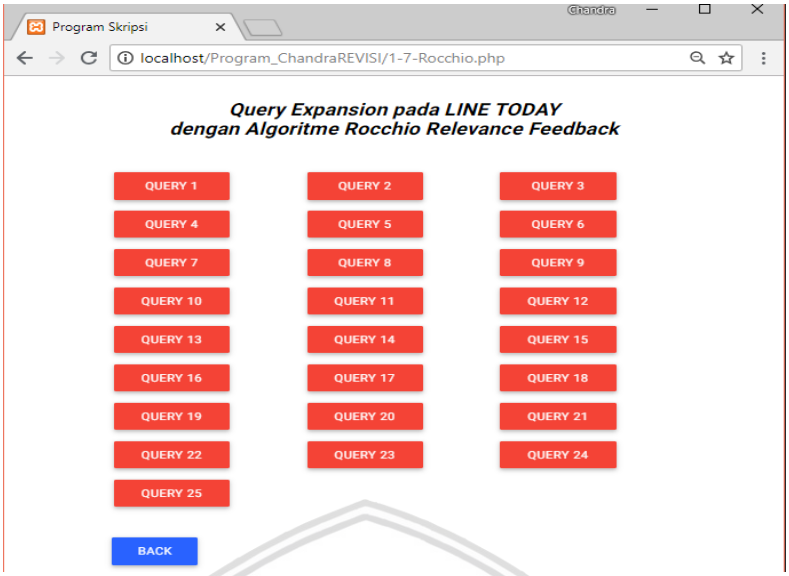
Pada halaman ini merupakan tampilan halaman dari dokumen-dokumen yang dihasilkan dari *query* yang diinputkan setelah ditambah dengan *query* baru. Tiap dokumen akan memiliki *checkbox* untuk penilaian yang diberikan oleh pengguna, dimana jika *checkbox* tersebut dicentang, maka menandakan dokumen tersebut relevan. Berikut ditunjukkan pada Gambar 4.24.



Gambar 4.24 Tampilan Halaman Hasil Pencarian *Query* Tambahan

4.10.5 Tampilan Halaman *Rocchio Relevance Feedback*

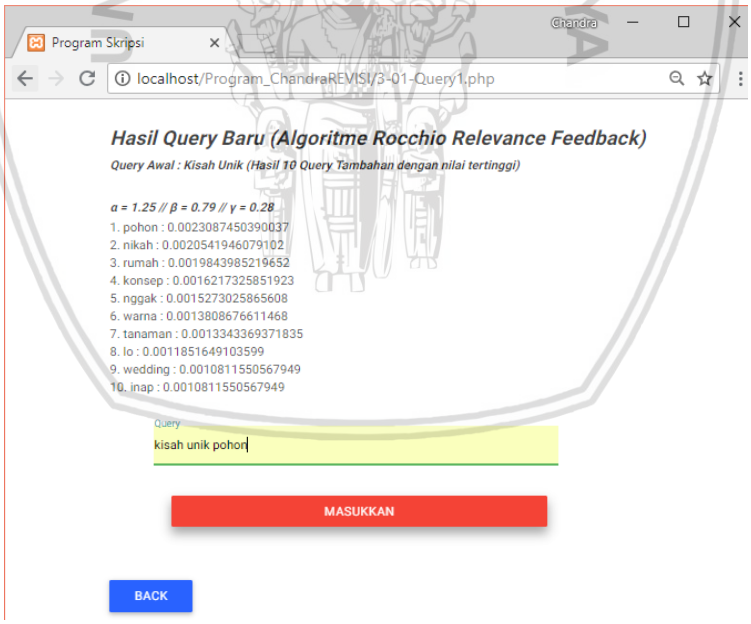
Pada bagian ini dilakukan pengujian pada algoritma tradisional dari *extended rocchio relevance feedback*, yaitu *rocchio relevance feedback*. Berikut tampilannya ditunjukkan pada Gambar 4.25.



Gambar 4.25 Tampilan Halaman *Rocchio Relevance Feedback*

4.10.5.1 Tampilan Halaman Hasil *Rocchio Relevance Feedback*

Berikut merupakan hasil *query* tambahan yang dihasilkan dari metode *Rocchio Relevance Feedback*, yang diurutkan berdasarkan 10 nilai teratas pada salah satu *query* dari 25 *query* uji yang ditunjukkan pada Gambar 4.26.



Gambar 4.26 Tampilan Halaman Hasil *Rocchio Relevance Feedback*

4.10.5.2 Tampilan Halaman Hasil Pencarian *Rocchio Relevance Feedback*

Pada tampilan halaman ini menghasilkan hasil dokumen-dokumen yang ditampilkan dari *query* pencarian yang diinputkan, yaitu terdiri dari *query* asli dan telah dikombinasi dengan *query* tambahan. Berikut tampilannya ditunjukkan pada Gambar 4.27.

Gambar 4.27 Tampilan Halaman Hasil Pencarian *Rocchio Relevance Feedback*

4.11 Penarikan Kesimpulan

Kesimpulan yang didapatkan berasal dari hasil dan uji keseluruhan dari mulai tahap perancangan, implementasi, hingga pengujian yang dilakukan. Penarikan kesimpulan akan didapat dari hasil analisis pengujian. Kesimpulan yang didapatkan nantinya bias digunakan untuk saran yang diharapkan untuk dapat memperbaiki kesalahan atau kekurangan pada penelitian kedepannya.

BAB 5 PENGUJIAN DAN ANALISIS

5.1 Pengujian

Pada pengujian ini dilakukan dengan menguji parameter σ, α, β, γ, dan Δ terlebih dahulu pada salah satu *query,* yaitu 'Avenger Infinity War'. Tiap parameter akan diuji dengan indikator *precision, recall,* dan *f-measure*, guna menemukan nilai terbaik, dimana tiap parameter akan dinilai nilai kenaikannya dan tiap parameter dengan kenaikan tertinggi, akan dipakai untuk menguji nilai kenaikan tambahan pada keseluruhan *query.* Berikut detail pengujian parameternya yang ditujukan pada Tabel 5.1 hingga Tabel 5.5.

Pada Tabel 5.1 menunjukkan pengujian yang dilakukan pada parameter *tho*, dimana saat pengujian dilakukan, parameter *tho* diset dengan nilai acak sedangkan untuk parameter lainnya diset dengan nilai yang sama. Selanjutnya dilakukan perhitungan untuk mendapatkan nilai kenaikan. Nilai kenaikan didapat dengan mengurangkan nilai antara *Q Baru* dengan *Q Awal*. Nilai kenaikan tertinggi pada parameter *tho,* maka akan digunakan atau terpilih sebagai nilai *tho* yang digunakan untuk menguji semua *query*.

Tabel 5.1 Hasil Pengujian Parameter *Tho*

No	σ	α	β	γ	Δ	Q Awal			Q Baru			Kenaikan		
						P	R	F	P	R	F	P	R	F
1	0.1	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
2	0.2	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
3	0.25	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
4	0.3	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
5	0.35	1.25	0.79	0.28	0.54	0.6	1	0.75	0.66667	1	0.8	0.06667	0	0.05

Pada Tabel 5.2 menunjukkan pengujian yang dilakukan pada parameter *alpha*, dimana saat pengujian dilakukan, parameter *alpha* diset dengan nilai acak sedangkan untuk parameter lainnya diset dengan nilai yang sama. Pada nilai *tho* di set dengan menggunakan nilai *tho* sebelumnya dengan kenaikan tertinggi.

Tabel 5.2 Hasil Pengujian Parameter *Alpha*

No	σ	α	β	γ	Δ	Q Awal			Q Baru			Kenaikan		
						P	R	F	P	R	F	P	R	F
1	0.1	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
2	0.1	1.28	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
3	0.1	1.35	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
4	0.1	1	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
5	0.1	1.38	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211

Pada Tabel 5.3 menunjukkan pengujian yang dilakukan pada parameter *beta*, dimana saat pengujian dilakukan, parameter *beta* diset dengan nilai acak sedangkan untuk parameter lainnya diset dengan nilai yang sama. Pada nilai *tho*

dan *alpha* di set dengan menggunakan nilai sebelumnya dengan kenaikan tertinggi.

Tabel 5.3 Hasil Pengujian Parameter *Beta*

No	σ	α	β	γ	Δ	Q Awal			Q Baru			Kenaikan		
						P	R	F	P	R	F	P	R	F
1	0.1	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
2	0.1	1.25	0.83	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
3	0.1	1.25	0.85	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
4	0.1	1.25	0.87	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
5	0.1	1.25	0.9	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211

Pada Tabel 5.4 menunjukkan pengujian yang dilakukan pada parameter *gamma*, dimana saat pengujian dilakukan, parameter *gamma* diset dengan nilai acak sedangkan untuk parameter lainnya diset dengan nilai yang sama. Pada nilai *tho, alpha,* dan *beta* di set dengan menggunakan nilai sebelumnya dengan kenaikan tertinggi.

Tabel 5.4 Hasil Pengujian Parameter *Gamma*

No	σ	α	β	γ	Δ	Q Awal			Q Baru			Kenaikan		
						P	R	F	P	R	F	P	R	F
1	0.1	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
2	0.1	1.25	0.79	0.3	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
3	0.1	1.25	0.79	0.35	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
4	0.1	1.25	0.79	0.38	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
5	0.1	1.25	0.79	0.4	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211

Pada Tabel 5.5 menunjukkan pengujian yang dilakukan pada parameter *delta*, dimana saat pengujian dilakukan, parameter *delta* diset dengan nilai acak sedangkan untuk parameter lainnya diset dengan nilai yang sama. Pada nilai *tho, alpha, beta,* dan *gamma* di set dengan menggunakan nilai sebelumnya dengan kenaikan tertinggi.

Tabel 5.5 Hasil Pengujian Parameter *Delta*

No	σ	α	β	γ	Δ	Q Awal			Q Baru			Kenaikan		
						P	R	F	P	R	F	P	R	F
1	0.1	1.25	0.79	0.28	0.54	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
2	0.1	1.25	0.79	0.28	0.57	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
3	0.1	1.25	0.79	0.28	0.6	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
4	0.1	1.25	0.79	0.28	0.66	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211
5	0.1	1.25	0.79	0.28	0.7	0.6	1	0.75	0.72727	1	0.84211	0.12727	0	0.09211

Parameter yang terpilih didapat dengan nilai kenaikan tertinggi di tiap pengujian parameter, namun nilai tiap parameter memiliki nilai yang cenderung sama, sehingga dipilih nilai pada no. 1 di tiap pengujian. Pada Tabel 5.6 menunjukkan nilai-nilai parameter hasil dari pengujian di atas, yang terpilih untuk dilakukan pengujian pada keseluruhan *query*, yaitu sebanyak 25 *query.*

Tabel 5.6 Nilai Parameter Terpilih

σ	α	β	γ	Δ
0.1	1.25	0.79	0.28	0.54

5.1.1 Skenario Pengujian 1

Pada pengujian skenario 1 ini dilakukan dengan menguji ke-25 *query* dengan menggunakan parameter yang sebelumnya telah diuji, dimana parameter yang terpilih, ialah parameter dengan nilai kenaikan tertinggi. Skenario ini menguji nilai kenaikan antara *query* asli dengan *query* yang ditambahkan dengan 1 kata tambahan. Berikut merupakan skenario pengujian 1 pada *query* dengan ditambahkan 1 kata dari *query* asli yang ditunjukkan pada Tabel 5.7.

Tabel 5.7 Skenario Pengujian 1

No	Indikator	*Query*		Kenaikan
		Awal	Baru	
1	***Precision***	0.26087	0.41026	0.14939
	Recall	0.54546	0.76191	0.21645
	F-Measure	0.35294	0.53333	0.18039
2	***Precision***	0.35714	0.5	0.14286
	Recall	0.38462	0.5	0.11539
	F-Measure	0.37037	0.5	0.12963
3	***Precision***	0.83333	0.71429	-0.11905
	Recall	0.22727	0.23810	0.01082
	F-Measure	0.35714	0.35714	0
4	***Precision***	0.33333	0.5	0.16667
	Recall	1	1	0
	F-Measure	0.5	0.66667	0.16667
5	***Precision***	0.29412	0.44444	0.15033
	Recall	1	1	0
	F-Measure	0.45455	0.61539	0.16083
6	***Precision***	0.83333	0.76923	-0.0641
	Recall	1	1	0
	F-Measure	0.90909	0.86957	-0.03953
7	***Precision***	0.6	0.72727	0.12727
	Recall	1	1	0
	F-Measure	0.75	0.84211	0.09211
8	***Precision***	0.73333	0.75	0.01667
	Recall	0.73333	0.75	0.01667
	F-Measure	0.73333	0.75	0.01667
9	***Precision***	0.69231	0.45	-0.24231
	Recall	0.5625	0.5625	0

Tabel 5.7 Skenario Pengujian 1 (lanjutan)

No	Indikator	*Query*		Kenaikan
		Awal	Baru	
9	***F-Measure***	0.62069	0.5	-0.12069
10	***Precision***	0.28571	0.27778	-0.00794
	Recall	1	1	0
	F-Measure	0.44444	0.43478	-0.00966
11	***Precision***	0.5	0.66667	0.16667
	Recall	0.8	0.85714	0.05714
	F-Measure	0.61539	0.75	0.13462
12	***Precision***	0.5	0.77778	0.27778
	Recall	0.66667	0.77778	0.11111
	F-Measure	0.57143	0.77778	0.20635
13	***Precision***	0.625	0.66667	0.04167
	Recall	0.41667	0.46154	0.04487
	F-Measure	0.5	0.54546	0.04546
14	***Precision***	0.41667	0.53333	0.11667
	Recall	0.5	0.66667	0.16667
	F-Measure	0.45455	0.59259	0.13805
15	***Precision***	0.6	0.57143	-0.0286
	Recall	0.6	0.66667	0.06667
	F-Measure	0.6	0.61539	0.01539
16	***Precision***	0.3125	0.3125	0
	Recall	1	1	0
	F-Measure	0.47619	0.47619	0
17	***Precision***	0.77778	0.11111	-0.6667
	Recall	0.875	1	0.125
	F-Measure	0.82353	0.2	-0.62353
18	***Precision***	0.58333	0.3125	-0.27083
	Recall	0.7	0.83333	0.13333
	F-Measure	0.63636	0.45455	-0.18182
19	***Precision***	0.66667	0.85714	0.19048
	Recall	0.66667	0.75	0.08333
	F-Measure	0.66667	0.8	0.13333
20	***Precision***	0.66667	0.28571	-0.3810
	Recall	1	1	0
	F-Measure	0.8	0.44444	-0.35556
21	***Precision***	0.85714	0.85714	0
	Recall	1	1	0
	F-Measure	0.92308	0.92308	0
22	***Precision***	0.66667	0.16667	-0.5

Tabel 5.7 Skenario Pengujian 1 (lanjutan)

No	Indikator	*Query*		Kenaikan
		Awal	Baru	
22	***Recall***	1	1	0
	F-Measure	0.8	0.28571	-0.51429
23	***Precision***	0.75	0.33333	-0.41667
	Recall	0.16667	1	0.83333
	F-Measure	0.27273	0.5	0.22727
24	***Precision***	0.66667	0.76923	0.10256
	Recall	0.88889	0.90909	0.02020
	F-Measure	0.76191	0.83333	0.07143
25	***Precision***	0.66667	0.5625	-0.10417
	Recall	0.44444	0.69231	0.24786
	F-Measure	0.53333	0.62069	0.08736

Berdasarkan tabel di atas, diperoleh rata-rata kenaikan pada tiap *precision, recall,* dan *f-measure*. Rata-rata kenaikan didapat dengan menghitung nilai rata-rata di tiap masing-masing indicator. Nilai minus mengartikan bahwa tidak terjadi kenaikan antara *query* awal dengan *query* baru, melainkan terjadi penurunan. Beikut rata-rata kenaikan yang ditunjukkan pada Tabel 5.8.

Tabel 5.8 Rata-Rata Kenaikan

Precision	-0.0461
Recall	0.08995
F-Measure	-0.0016

5.1.2 Skenario Pengujian 2

Pada skenario pengujian 2 ini membandingkan nilai akurasi dari *query* dengan tambahan kata sebanyak 1 dan *query* dengan tambahan kata sebanyak 2. Nilai akurasi dari *query* dengan tambahan 1 kata cenderung lebih tinggi dibandingkan dengan nilai akurasi dari *query* dengan tambahan 2 kata. Hal ini disebabkan semakin banyak *query* atau kata yang diinputkan, maka data yang ditampilkan akan semakin spesifik, yang menyebabkan dokumen relevan akan berkurang dan dokumen yang tidak relevan akan semakin bertambah. Berikut merupakan pengujian pada kata yang ditambahkan dalam *query* yang ditunjukkan pada Tabel 5.9.

Tabel 5.9 Pengujian Kata Tambahan

No	1 Kata				2 Kata			
	Indikator			Akurasi	Indikator			Akurasi
	Precision	*Recall*	*F-Measure*		*Precision*	*Recall*	*F-Measure*	
1	0.41026	0.7619	0.53333	0.86	0.39535	0.85	0.53968	0.855
2	0.5	0.5	0.5	0.93	0.4	0.5	0.44444	0.925
3	0.71429	0.2381	0.35714	0.91	0.71429	0.23810	0.35714	0.91
4	0.5	1	0.66667	0.99	0.5	1	0.66667	0.985
5	0.44444	1	0.61538	0.95	0.05556	1	0.10526	0.915
6	0.76923	1	0.86957	0.985	0.76923	1	0.86957	0.985
7	0.72727	1	0.84211	0.985	0.38462	1	0.55556	0.96
8	0.75	0.75	0.75	0.96	0.75	0.75	0.75	0.96
9	0.45	0.5625	0.5	0.91	0.42858	0.5625	0.48649	0.905
10	0.27778	1	0.43478	0.935	0.26316	1	0.41667	0.93
11	0.66667	0.85714	0.75	0.98	0.66667	0.85714	0.75	0.98
12	0.77778	0.77778	0.77778	0.98	0.55556	0.71429	0.625	0.97
13	0.66667	0.46154	0.54545	0.95	0.7	0.53846	0.60870	0.955
14	0.53333	0.66667	0.59259	0.945	0.5	0.66667	0.57143	0.94
15	0.57143	0.66667	0.61538	0.975	0.57143	0.66667	0.61538	0.975
16	0.3125	1	0.47619	0.945	0.27778	1	0.43478	0.935
17	0.11111	1	0.2	0.96	0.05	1	0.09524	0.905
18	0.3125	0.83333	0.45455	0.94	0.29412	0.83333	0.43478	0.935
19	0.85714	0.75	0.8	0.985	0.625	0.71429	0.66667	0.975
20	0.28571	1	0.44444	0.975	0.25	1	0.4	0.97
21	0.85714	1	0.92308	0.995	0.75	1	0.85714	0.99
22	0.16667	1	0.28571	0.975	0.11111	1	0.2	0.96
23	0.33333	1	0.5	0.99	0.25	1	0.4	0.985
24	0.76923	0.90909	0.83333	0.98	0.52632	0.90909	0.66667	0.95
25	0.5625	0.69231	0.62069	0.945	0.52941	0.69231	0.6	0.94

Dari percobaan yang dilakukan pada Tabel 5.9, maka diperoleh nilai rata-rata seperti yang ditunjukkan pada Tabel 5.10.

Tabel 5.10 Skenario Pengujian 2

Jumlah Kata	*Precision*	*Recall*	*F-Measure*	Akurasi
1 kata	0.53308	0.81708	0.59553	0.9574
2 kata	0.45273	0.81971	0.52469	0.9478

5.1.3 Skenario Pengujian 3

Pada skenario ini dilakukan pengujian pada *Precision@K,* dimana digunakan untuk mengukur *threshold* pada peringkat K. Dokumen yang dihitung ialah

dokumen relevan teratas sejumlah *K* dan mengabaikan dokumen peringkat yang berada di bawah *K.* Berikut pengujian pada *Precision@K* yang ditunjukkan pada Tabel 5.11.

Tabel 5.11 Skenario Pengujian 3

No	Query	Peringkat	Ket.	P@10	Tambahan Kata
1	Kisah Unik **Akun**	**1**	R	0.2	1 kata
		2	T		
		3	T		
		4	T		
		5	T		
		6	T		
		7	R		
		8	T		
		9	T		
		10	T		
2	Kisah Unik **Akun Artinstic**	**1**	R	0.3	2 kata
		2	T		
		3	T		
		4	R		
		5	T		
		6	T		
		7	R		
		8	T		
		9	T		
		10	T		
3	Kisah Unik **Akun Artinstric Angkat**	**1**	R	0.4	3 kata
		2	T		
		3	T		
		4	T		
		5	R		
		6	R		
		7	T		
		8	R		
		9	T		
		10	T		
4	Kisah Unik **Akun Artinstric Angkat Blogger**	**1**	R	0.5	4 kata
		2	T		
		3	R		
		4	T		
		5	T		
		6	T		
		7	R		
		8	R		
		9	T		
		10	R		

Tabel 5.11 Skenario Pengujian 3 (lanjutan)

No	Query	Peringkat	Ket.	P@10	Tambahan Kata
5	Kisah Unik **Akun Artinstric Angkat Blogger Catching**	**1**	R	0.6	5 kata
		2	R		
		3	R		
		4	T		
		5	T		
		6	T		
		7	R		
		8	R		
		9	T		
		10	R		
6	Kisah Unik **Akun Artinstric Angkat Blogger Catching Busana**	**1**	R	0.5	6 kata
		2	R		
		3	T		
		4	T		
		5	R		
		6	T		
		7	R		
		8	T		
		9	T		
		10	R		
7	Kisah Unik **Akun Artinstric Angkat Blogger Catching Busana Code**	**1**	R	0.6	7 kata
		2	R		
		3	R		
		4	T		
		5	T		
		6	R		
		7	R		
		8	R		
		9	T		
		10	T		
8	Kisah Unik **Akun Artinstric Angkat Blogger Catching Busana Code Banding**	**1**	R	0.6	8 kata
		2	R		
		3	R		
		4	T		
		5	R		
		6	T		
		7	R		
		8	R		
		9	T		
		10	T		
9	Kisah Unik **Akun Artinstric Angkat Blogger Catching Busana Code Banding Eye**	**1**	R	0.8	9 kata
		2	R		
		3	R		

Tabel 5.11 Skenario Pengujian 3 (lanjutan)

No	Query	Peringkat	Ket.	P@10	Tambahan Kata
9	Kisah Unik **Akun Artinstric Angkat Blogger Catching Busana Code Banding Eye**	**4**	R	0.8	9 kata
		5	R		
		6	R		
		7	T		
		8	R		
		9	R		
		10	T		
10	Kisah Unik **Akun Artinstric Angkat Blogger Catching Busana Code Banding Eye Garis**	**1**	R	0.9	10 kata
		2	R		
		3	R		
		4	R		
		5	R		
		6	R		
		7	R		
		8	T		
		9	R		
		10	R		

5.1.4 Skenario Pengujian 4

Pada skenario pengujian ini dilakukan untuk mengukur dan membandingkan metode yang digunakan pada penelitian ini, yaitu *Extended Rocchio Relevance Feedback,* dengan metode sebelumnya, yaitu *Rocchio Relevance Feedback*. Pengujian ini dilakukan pada tambahan 1 kata dari *query* asli. Berikut merupakan perbandingannya ditunjukkan pada Tabel 5.12.

Tabel 5.12 Perbandingan Metode

No	*Rocchio Relevance Feedback*				*Extended Rocchio Relevance Feedback*			
	Indikator			Akurasi	Indikator			Akurasi
	Precision	*Recall*	*F-Measure*		*Precision*	*Recall*	*F-Measure*	
1	0.08333	0.66667	0.14815	0.885	0.41026	0.7619	0.53333	0.86
2	0.44444	1	0.61539	0.95	0.5	0.5	0.5	0.93
3	0.03448	1	0.06667	0.86	0.71429	0.2381	0.35714	0.91
4	0.33333	1	0.5	0.99	0.5	1	0.66667	0.99
5	0	0	0	0.89	0.44444	1	0.61538	0.95
6	0.30769	1	0.47059	0.955	0.76923	1	0.86957	0.985
7	0.42857	1	0.6	0.96	0.72727	1	0.84211	0.985
8	0.29412	1	0.45455	0.94	0.75	0.75	0.75	0.96
9	0.06667	1	0.125	0.93	0.45	0.5625	0.5	0.91
10	0.18182	1	0.30769	0.955	0.27778	1	0.43478	0.935
11	0.125	1	0.22222	0.965	0.66667	0.85714	0.75	0.98

Tabel 5.12 Perbandingan Metode (lanjutan)

No	*Rocchio Relevance Feedback*				*Extended Rocchio Relevance Feedback*			
	Indikator			Akurasi	Indikator			Akurasi
	Precision	*Recall*	*F-Measure*		*Precision*	*Recall*	*F-Measure*	
12	0.15556	1	0.26923	0.81	0.77778	0.77778	0.77778	0.98
13	0.875	0.5	0.63636	0.96	0.66667	0.46154	0.54545	0.95
14	0.05882	1	0.11111	0.84	0.53333	0.66667	0.59259	0.945
15	0.57143	1	0.72727	0.985	0.57143	0.66667	0.61538	0.975
16	0.04348	1	0.08333	0.89	0.3125	1	0.47619	0.945
17	0.1	1	0.18182	0.955	0.11111	1	0.2	0.96
18	0.06667	1	0.125	0.93	0.3125	0.83333	0.45455	0.94
19	0.5	1	0.66667	0.98	0.85714	0.75	0.8	0.985
20	0.66667	1	0.8	0.995	0.28571	1	0.44444	0.975
21	0.16667	1	0.28571	0.95	0.85714	1	0.92308	0.995
22	0.14286	1	0.25	0.97	0.16667	1	0.28571	0.975
23	0.22222	0.5	0.30769	0.955	0.33333	1	0.5	0.99
24	0.33333	0.8	0.47059	0.955	0.76923	0.90909	0.83333	0.98
25	0.28571	1	0.44444	0.975	0.5625	0.69231	0.62069	0.945

Berikut merupakan rata-rata pengujiannya yang dilakukan dengan indikator *Precision, Recall, F-Measure,* dan Akurasinya, yang ditunjukkan pada Tabel 5.13.

Tabel 5.13 Skenario Pengujian 4

Metode	*Precision*	*Recall*	*F-Measure*	Akurasi
Extended Rocchio	0.53308	0.81708	0.59553	0.9574
Rocchio	0.25951	0.89867	0.35478	0.9372

5.2 Analisis

Dari pengujian skenario 1, rata-rata nilai kenaikan pada *precision* mengalami penurunan, karena nilai FP mengalami kenaikan, dimana FP atau *false positive*, merupakan banyaknya dokumen yang tidak sesuai dengan *query*. Sedangkan pada *recall* cenderung naik, hal ini terjadi, karena nilai FN atau *false negative* yang cenderung menurun. Nilai FN ditujukan pada banyaknya dokumen relevan yang tidak masuk dalam perangkingan, hal ini menunjukkan bahwa dokumen relevan yang tidak masuk dalam perangkingan cenderung sedikit. Sedangkan untuk *f-measure,* merupakan kombinasi dari hasil *precision* dan *recall.*

Pada pengujian skenario 2, terbukti bahwa dengan penambahan kata menjadi 2 tambahan kata, maka nilai *precision* akan menurun, dikarenakan banyak dokumen yang tidak sesuai muncul dalam perangkingan. Nilai *recall* cenderung naik, karena dok relevan yang tidak muncul dalam perangkingan cenderung sedikit. Pada nilai akurasi mengalami penurunan dari tambahan 1 kata menjadi 2

kata. Hal ini disebabkan semakin banyaknya *query* yang diinputkan, maka dokumen yang masuk dalam perangkingan akan semakin meningkat, namun akan semakin sedikit banyaknya dokumen yang relevan dengan *query* dan dokumen yang tidak relevan akan semakin bertambah. Semakin banyak *query*, maka semakin spesifik *query* pencarian yang mengakibatkan banyaknya dokumen yang relevan dengan *query* juga semakin terbatas.

Untuk pengujian skenario 3 dilakukan pengujian pada *threshold* nilai perangkingan K=10. Pengujian ini dilakukan untuk menguji nilai *precision*, dimana peringkat perangkingan teratas dari K akan dihitung dan mengabaikan nilai peringkat perangkingan dibawah K. Pada Tambahan kata ke-6 terjadi penurunan pada nilai *P@K*, karena dokumen relevan mengalami penurunan, sedangkan pada tambahan kata lainnya cenderung tetap dan naik.

Pengujian skenario 4 dilakukan perbandingan antara metode penelitian ini, yaitu *Extended Rocchio Relevance Feedback* dengan metode sebelumnya, yaitu *Rocchio Relevance Feedback.* Pada tiap indikator *precision, recall, f-measure,* dan akurasinya cenderung tinggi pada metode *Extended Rocchio Relevance Feedback,* hal ini dikarenakan adanya feedback dari penilaian pengguna pada dokumen relevan dan tidak relevan, yang mengakibatkan tiap *query expansion* yang dihasilkan memiliki dokumen relevan yang sesuai.

BAB 6 PENUTUP

6.1 Kesimpulan

Pada bagian ini akan membahas tentang kesimpulan yang didapatkan dari hasil penelitian *Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback,* berikut diantaranya:

1. Metode *Extended Rocchio Relevance Feedback* dapat diterapkan dalam melakukan pencarian pada situs berita online LINE TODAY. Dokumen yang tersedia akan melalui tahapan *preprocessing*, kemudian dilakukan perhitungan pada *term weighting* dan *cosine similarity*, selanjutnya dilakukan pencarian *Term Vector P, N,* dan *F,* hingga set parameter untuk dapat digunakan dalam perhitungan untuk pencarian *query* tambahan baru. Semakin banyak *query* yang ditambahkan dari *query* aslinya, maka pencarian akan semakin spesifik dan dokumen yang relevanpun akan semakin sedikit sebaliknya dokumen tidak relevan akan semakin banyak.
2. Pengujan *query expansion* dengan Metode *Extended Rocchio Relevance Feedback* menghasilkan nilai rata-rata *precision* sebesar 0.53 untuk 1 kata tambahan dan 0.45 untuk 2 tambahan kata. Untuk nilai *recall* pada 1 tambahan kata sebesar 0.817 dan 0.819 untuk 2 tambahan kata. Kemudian pada *f-measure* memiliki nilai rata-rata sebesar 0.596 untuk 1 kata tambahan dan 0.525 untuk 2 tambahan kata. Sedangkan rata-rata akurasinya pada 1 kata tambahan sebesar 0.96 dan 0.95 untuk 2 tambahan kata. Penambahan jumlah kata pada *query expansion* dapat mempengaruhi nilai *precision, recall, f-measure,* dan akurasinya. Selain itu dilakukan perbandingan dengan metode tradisional, yaitu *Rocchio* dan hasilnya terbukti lebih baik dengan menggunakan *Extended Rocchio* dengan kenaikan hingga 2%.

6.2 Saran

Berikut merupakan saran dari penulis yang didapat dari hasil penelirian *Query Expansion* Pada LINE TODAY Dengan Algoritme *Extended Rocchio Relevance Feedback* untuk pengembangan lebih lanjut, diantaranya:

1 Pengembangan metode dilakukan dengan metode lain, seperti semantik yang dapat meningkatkan pemilihan kata-kata tambahan sehingga dapat terangkai menjadi *query* yang semakin baik dan memiliki kesamaan yang tinggi dibanding dengan *query* awal dari pengguna.

2 Untuk penelitian selanjutnya memanfaatkan teknik *relevance feedback* dengan berdasarkan pada pendekatan lain, guna untuk memperluas dan menguji kemampuan *query expansion*.

DAFTAR PUSTAKA

Adisantoso, J., Ridha, A., & Agusetyawan, W. (2006). Relevance feedback pada temu-kembali teks berbahasa indonesia dengan metode ide-dec-hi dan ide-regular. *Jurnal Ilmiah Ilmu Komputer*, 1-8.

Alam, M., & Sadaf, K. (2015). Relevance feedback versus web search document clustering. *IEEE Conference* (pp. 4294-4298). India: BharatiVidyapeeth's Institute of Computer Applications and Management (BVICAM).

Blair, David C. (2003). *Information retrieval and the philosophy of language.* New York: Elseiver North-Holland, Inc. New York, NY, USA.

Fauzi, M. A., Arifin, A. Z., & Yuniarti, A. (2014). Term weighting berbasis indeks buku dan kelas untuk perangkingan dokumen berbahasa arab. *LONTAR KOMPUTER VOL. 5, NO.2*, 435-442.

Hamzah, Amir. (2006). Pengaruh stemming kata dalam peningkatan unjuk kerja document clustering untuk dokumen berbahasa indonesia. *Seminar Nasional Riset Teknologi Informasi-SRITI 2006* (pp. 253-263). Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Komputer AKAKDM Yogyakarta.

Hazimeh, H., & Zhai, C. (2015). Axiomatianalysis of smoothing methods in language models for pseudo-relevance feedback. *ICTIR '15 Proceedings of the 2015 International Conference on The Theory of Information Retrieval* (pp. 141-150). Northampton, Massachusetts, USA: ACM New York, NY, USA.

Herlambang, Y. R., Putri, R. R., & Wihandika, R. C. (2017). Implementasi metode k-nearest neighbour dengan pembobotan tf.idf.icf untuk kategorisasi ide kreatif pada perusaHAAN. *JTIIK*, 97-103.

Jordan, C., & Watters, C. (2004). Extending the rocchio relevance feedback algorithm to provide contextual retrieval. *International AWIC 2004: Advances in Web Intelligence* (pp. 135-144). Berlin, Heidelberg: Springer.

Kurniawan, B., Effendi, S., & Sitompul, O. S. (2012). Klasifikasi konten berita dengan metode text mining. *Jurnal Dunia Teknologi Informasi Vol.1, No.1*, 14-19.

Ludviani, R., Hayati, K. F., Arifin, A. Z., & Purwitasari, D. (2015). Optimasi pembobotan pada query expansion dengan term relatedness to Query-Entropy based (TRQE). *Jurnal Buana Informatika, Volume 6, Nomor 3*, 203-212.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge: Cambridge University Press.

Nata, G. N., & Yudiastra, P. P. (2017). Preprocessing text mining pada email box berbahasa indonesia. *Konferensi Nasional Sistem & Informatika* (pp. 479-483). Bali: STMIK STIKOM.

Pamungkas, Z. Y., Indriati, & Ridok, A. (2015). Query expansion pada sistem temu kembali informasi dokumen berbahasa indonesia menggunakan pseudo relevance feedback. *Program Teknologi Informatika dan Ilmu Komputer Universitas Brawijaya*.

Rosid, M. A., Gunawan, & Pramana, E. (2015). Centroid based classifier dengan fitur tf-idf-icf untuk klasifikasi keluhan mahasiswa pada aplikasi e-complaint di universitas muhammadiyah sidoarjo. *jTE-U, Vol. 1, No. 1*, 1-7.

Rustiana, D., & Rahayu, N. (2017). Analisis sentimen pasar otomotif mobil: tweet twitter menggunakan naïve bayes. *Jurnal SIMETRIS, Vol 8 No 1*, 113-120.

Saneifar, H., Bonniol, S., Poncelet, P., & Roche, M. (2014). Enchacing passage retrieval in log files by query expansion based on explicit and pseudo relevance feedback. *H. Saneifer, et al*, 1-15.

Vijayarani, S., & Janani, R. (2016). Text mining: open source tokenization tools-an analysis. *advanced computational intelligence: An International Journal (ACII)), Vol.3, No.1*, 37-47.

Wahyudi, D., Susyanto, T., & Nugroho, D. (2017). Implementasi dan analisis algoritma stemming nazief& adriani dan porter pada dokumen berbahasa indonesia. *SINUS*, 49-56.

Yates, R. B., & Neto, B. R. (1999). *Modern information retrieval.* New York: ACM Press.

Yugianus, P., Dachlan, S. H., & Hasanah, R. N. (2013). Pengembangan sistem penelusuran katalog perpustakaan dengan metode rocchio relevance feedback. *Jurnal EECCIS Vol. 7, No. 1*, 47-52.